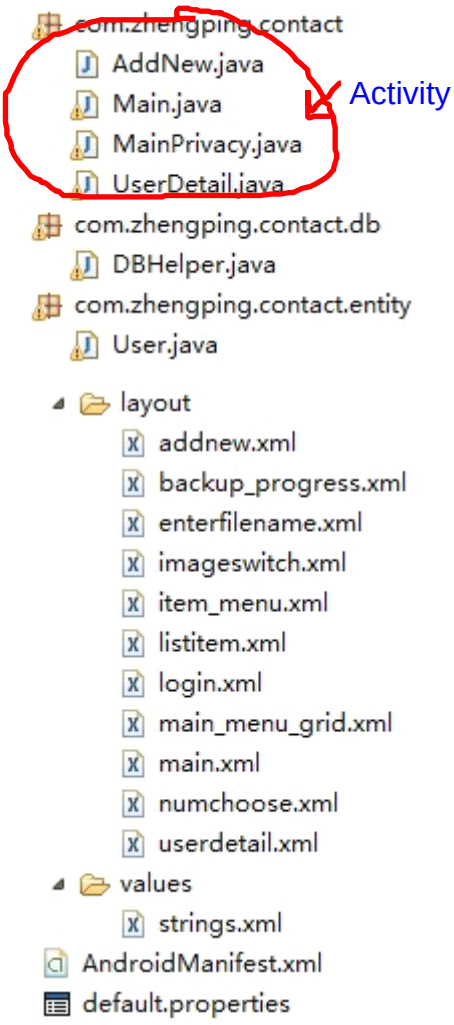
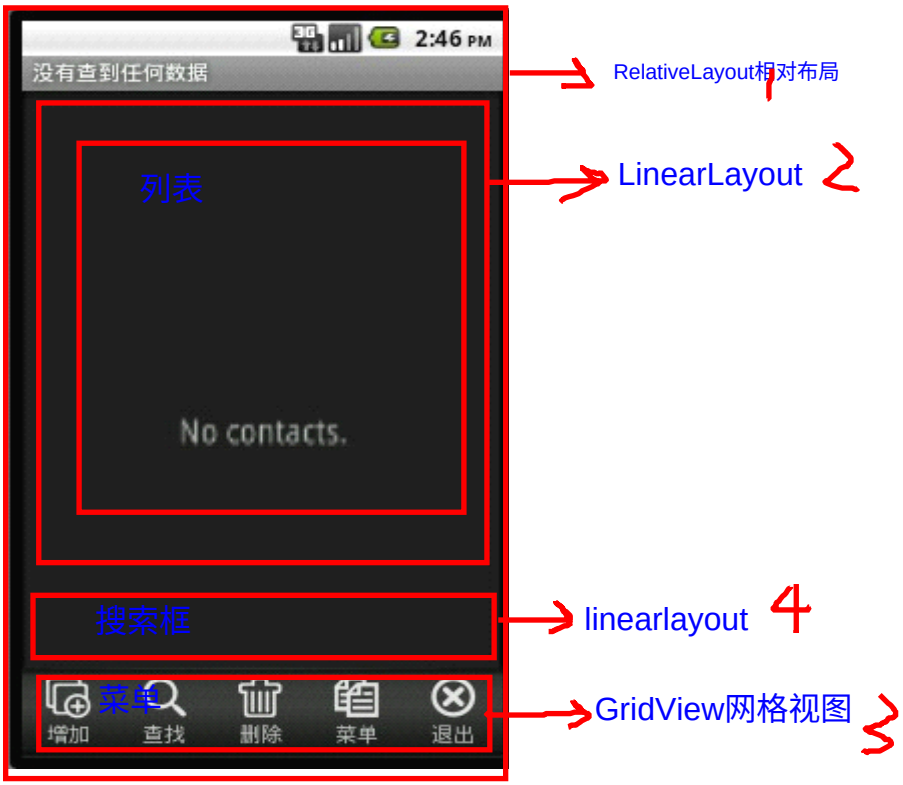
目录

1. 目录结构



2. 界面设计

2.1 Main.xml




```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 相对布局
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout
        android:id="@+id/list_ll"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
    <ListView   列表
        android:id="@+id/lv_userlist"
        android:layout_above="@+id/gv_buttom_menu"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
    </ListView>
    </LinearLayout>
    <GridView   网格视图
        android:id="@+id/gv_buttom_menu"
        android:layout_height="65sp"
```

```
        android:layout_width="fill_parent"
        android:layout_alignParentBottom="true"
        android:visibility="gone">
    </GridView>
<LinearLayout
        android:id="@+id/ll_search"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_alignParentLeft="true"
        android:layout_above="@+id/gv_buttom_menu"
        android:visibility="gone">
        <EditText
            android:id="@+id/et_search"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:hint="Enter the name or phone"
            android:textSize="18sp">
        </EditText>
</LinearLayout>
</RelativeLayout>
```

位于菜单上方

搜索框

2.2 main_menu_grid.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
<GridView
        android:id="@+id/gridview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:numColumns="3"
        android:verticalSpacing="10dip"
        android:horizontalSpacing="10dip"
        android:stretchMode="columnWidth"
        android:gravity="center"
        />
 </LinearLayout>
```

2.3 addnew.xml


```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" 与屏幕等宽
    android:layout_height="fill_parent" 与屏幕等高
    android:orientation="vertical"> 垂直
    <ScrollView android:layout_weight="4" 比重
                android:id="@+id/ScrollView1"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content" 根据内容决定高度
                android:scrollbars="vertical"> 滚动条方向-垂直方向
        <LinearLayout
            xmlns:android="http://schemas.android.com/apk/res/android"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <LinearLayout
            android:id="@+id/widget205"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            >
```

图片按钮

```
<ImageButton
    android:id="@+id/image_button"
        android:layout_width="60px"
        android:layout_height="60px"
        android:src="@drawable/icon"
        android:scaleType="centerCrop">
        </ImageButton>
        <EditText
            android:id="@+id/username"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:scrollbars="vertical"
            android:hint="姓名"
            android:gravity="top"
            android:layout_gravity="center_vertical"
        />
4</LinearLayout>
<LinearLayout
5       android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_marginRight="10dp"
        android:layout_marginLeft="10dp">
        <TextView
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:text="手机："
            android:textSize="20dp" />
        <EditText
            android:id="@+id/mobilephone"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:scrollbars="vertical"
            android:gravity="top" android:hint="手机"
            android:phoneNumber="true" />
5</LinearLayout>
6   <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp">
        <TextView
            android:layout_height="wrap_content"
```

按比例统一缩放图片
（保持图片的尺寸比例）
便于图片的两维（宽度和高度）
等于或大于相应的视图维度

hint:当文本内容为空时，出现的提示信息

通过设置android:gravity="center"
来让EditText中的文字在EditText组件中居中显示；
同时我们设置EditText的android:layout_gravity="right"
来让EditText组件在LinearLayout中居中显示

区别在于：android:gravity用于设置View组件的对齐方式，
而android:layout_gravity用于设置Container组件的对齐方式。

只能输入数字

```
            android:layout_width="wrap_content"
            android:text="办公室电话："
            android:textSize="20dp" />
        <EditText
            android:id="@+id/officephone"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:scrollbars="vertical"
            android:phoneNumber="true"
            android:hint="办公室电话"
            android:gravity="top" />
    </LinearLayout>
    <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp">
        <TextView
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:text="家庭电话："
            android:textSize="20dp" />
        <EditText
            android:id="@+id/familyphone"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:scrollbars="vertical"
            android:phoneNumber="true"
            android:hint="家庭电话"
            android:gravity="top" />
    </LinearLayout>
    <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp">
        <TextView
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:text="职务职称："
            android:textSize="20dp" />
        <EditText
```

```
            android:id="@+id/position"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:scrollbars="vertical"
            android:hint="职务职称"
            android:gravity="top" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp">
        <TextView
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:text="单位名称："
            android:textSize="20dp" />
        <EditText
            android:id="@+id/company"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:scrollbars="vertical"
            android:hint="单位名称"
            android:gravity="top" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp">
        <TextView
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:text="地址："
            android:textSize="20dp" />
        <EditText
            android:id="@+id/address"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:scrollbars="vertical"
            android:hint="地址"
            android:gravity="top" />
```

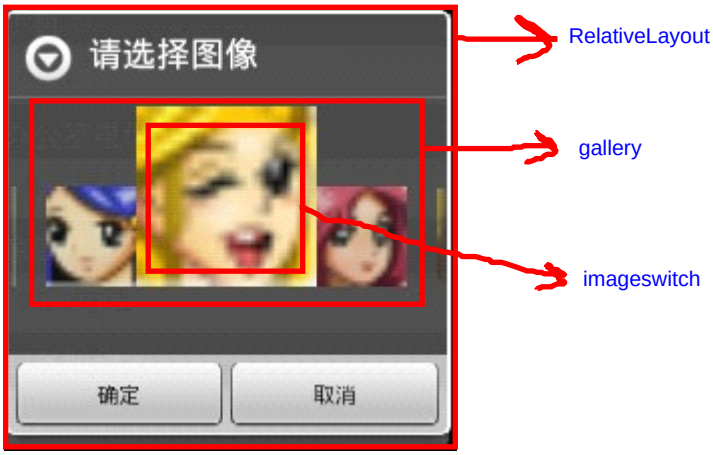
```
</LinearLayout>
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp">
    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="邮政编码："
        android:textSize="20dp" />
    <EditText
        android:id="@+id/zipcode"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:scrollbars="vertical"
        android:hint="邮政编码"
        android:gravity="top" />
</LinearLayout>
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp">
    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="E-mail："
        android:textSize="20dp" />
    <EditText
        android:id="@+id/email"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:scrollbars="vertical"
        android:hint="E-mail" android:gravity="top" />
</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_marginLeft="10dp"
```

```
            android:layout_marginRight="10dp">
            <TextView
                android:layout_height="wrap_content"
                android:layout_width="wrap_content"
                android:text="其他联系方式："
                android:textSize="20dp" />
            <EditText
                android:id="@+id/othercontact"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:scrollbars="vertical"
                android:hint="其他联系方式"
                android:gravity="top" />
        </LinearLayout>
        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp">
            <TextView
                android:layout_height="wrap_content"
                android:layout_width="wrap_content"
                android:text="备注："
                android:textSize="20dp" />
            <EditText android:id="@+id/remark"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:scrollbars="vertical"
                android:hint="备注" android:gravity="top" />
        </LinearLayout>
    </LinearLayout>
</ScrollView>
<LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp">
<Button
    android:id="@+id/save"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
```

```
        android:text="保存"/>
        <Button
        android:id="@+id/btn_return"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="返回"/>
        </LinearLayout>
</LinearLayout>
```

2.4 imageswitch.xml




```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
android:id="@+id/widget34"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
xmlns:android="http://schemas.android.com/apk/res/android"
>
<Gallery
android:id="@+id/gallery"
android:layout_width="fill_parent"
android:layout_height="110px"
android:layout_marginTop="10px"
android:layout_alignParentLeft="true"
>
</Gallery>
<ImageSwitcher
android:id="@+id/imageswitch"
android:layout_width="90px"
android:layout_height="90px"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"      水平居中
```

```
android:layout_alignBottom="@+id/gallery"
>
</ImageSwitcher>
</RelativeLayout>
```

2.5 item_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/RelativeLayout_Item"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:paddingBottom="5dip">
    <ImageView android:id="@+id/item_image"
         android:layout_centerHorizontal="true"
         android:layout_width="wrap_content"
          android:layout_height="wrap_content">
    </ImageView>
    <TextView
        android:layout_below="@id/item_image"
        android:id="@+id/item_text"
        android:layout_centerHorizontal="true"
        android:layout_width="wrap_content"
         android:layout_height="wrap_content">
    </TextView>
</RelativeLayout>
```

2.6 userdetail.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <ScrollView
        android:layout_weight="4"
        android:id="@+id/ScrollView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
       android:scrollbars="vertical">
           <LinearLayout
           xmlns:android="http://schemas.android.com/apk/res/android"
           android:layout_width="fill_parent"
           android:layout_height="wrap_content"
          android:orientation="vertical">
          <LinearLayout
              android:id="@+id/widget205"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:layout_marginLeft="10dp"
              android:layout_marginRight="10dp">
              <ImageButton
                  android:id="@+id/image_button"
                  android:layout_width="60px"
                  android:layout_height="60px"
                  android:src="@drawable/icon"
                  android:adjustViewBounds="false"
                  android:scaleType="centerCrop">
              </ImageButton>
              <EditText
                  android:id="@+id/username"
                  android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:scrollbars="vertical"
                  android:hint="姓名"
                  android:gravity="top"
                  android:layout_gravity="center_vertical"
                  />
          </LinearLayout>
          <LinearLayout
              android:layout_width="fill_parent"
               android:layout_height="wrap_content"
```

```
            android:orientation="vertical"
             android:layout_marginRight="10dp"
             android:layout_marginLeft="10dp">
            <TextView
                android:layout_height="wrap_content"
                android:layout_width="wrap_content"
                android:text="手机： "
                android:textSize="20dp" />
            <EditText
                android:id="@+id/mobilephone"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:scrollbars="vertical"
                android:gravity="top"
                android:hint="手机"
                android:phoneNumber="true" />
        </LinearLayout>
        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp">
            <TextView
                android:layout_height="wrap_content"
                android:layout_width="wrap_content"
                android:text="办公室电话： "
                android:textSize="20dp" />
            <EditText
                android:id="@+id/officephone"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:scrollbars="vertical"
                android:phoneNumber="true"
                android:hint="办公室电话"
                android:gravity="top" />
        </LinearLayout>
        <LinearLayout android:layout_width="fill_parent"
            android:layout_height="wrap_content"
android:orientation="vertical"
            android:layout_marginLeft="10dp"
android:layout_marginRight="10dp">
            <TextView android:layout_height="wrap_content"
                android:layout_width="wrap_content" android:text="家
```

```
庭电话："
                android:textSize="20dp" />
            <EditText android:id="@+id/familyphone"
                android:layout_width="fill_parent"
android:layout_height="wrap_content"
                android:scrollbars="vertical"
android:phoneNumber="true"
                android:hint="家庭电话" android:gravity="top" />
        </LinearLayout>
        <LinearLayout android:layout_width="fill_parent"
            android:layout_height="wrap_content"
android:orientation="vertical"
            android:layout_marginLeft="10dp"
android:layout_marginRight="10dp">
            <TextView android:layout_height="wrap_content"
                android:layout_width="wrap_content" android:text="职
务职称："
                android:textSize="20dp" />
            <EditText android:id="@+id/position"
android:layout_width="fill_parent"
                android:layout_height="wrap_content"
android:scrollbars="vertical"
                android:hint="职务职称" android:gravity="top" />
        </LinearLayout>
        <LinearLayout android:layout_width="fill_parent"
            android:layout_height="wrap_content"
android:orientation="vertical"
            android:layout_marginLeft="10dp"
android:layout_marginRight="10dp">
            <TextView android:layout_height="wrap_content"
                android:layout_width="wrap_content" android:text="单
位名称："
                android:textSize="20dp" />
            <EditText android:id="@+id/company"
android:layout_width="fill_parent"
                android:layout_height="wrap_content"
android:scrollbars="vertical"
                android:hint="单位名称" android:gravity="top" />
        </LinearLayout>
        <LinearLayout android:layout_width="fill_parent"
            android:layout_height="wrap_content"
android:orientation="vertical"
            android:layout_marginLeft="10dp"
android:layout_marginRight="10dp">
```

```
                <TextView android:layout_height="wrap_content"
                    android:layout_width="wrap_content" android:text="地
址："
                    android:textSize="20dp" />
                <EditText android:id="@+id/address"
android:layout_width="fill_parent"
                    android:layout_height="wrap_content"
android:scrollbars="vertical"
                    android:hint="地址" android:gravity="top" />
            </LinearLayout>
            <LinearLayout android:layout_width="fill_parent"
                android:layout_height="wrap_content"
android:orientation="vertical"
                android:layout_marginLeft="10dp"
android:layout_marginRight="10dp">
                <TextView android:layout_height="wrap_content"
                    android:layout_width="wrap_content" android:text="邮
政编码："
                    android:textSize="20dp" />
                <EditText android:id="@+id/zipcode"
android:layout_width="fill_parent"
                    android:layout_height="wrap_content"
android:scrollbars="vertical"
                    android:hint="邮政编码" android:gravity="top" />
            </LinearLayout>
            <LinearLayout android:layout_width="fill_parent"
                android:layout_height="wrap_content"
android:orientation="vertical"
                android:layout_marginLeft="10dp"
android:layout_marginRight="10dp">
                <TextView android:layout_height="wrap_content"
                    android:layout_width="wrap_content"
android:text="E-mail："
                    android:textSize="20dp" />
                <EditText android:id="@+id/email"
android:layout_width="fill_parent"
                    android:layout_height="wrap_content"
android:scrollbars="vertical"
                    android:hint="E-mail" android:gravity="top" />
            </LinearLayout>
            <LinearLayout android:layout_width="fill_parent"
                android:layout_height="wrap_content"
android:orientation="vertical"
                android:layout_marginLeft="10dp"
```

```
android:layout_marginRight="10dp">
                <TextView android:layout_height="wrap_content"
                    android:layout_width="wrap_content" android:text="其
他联系方式："
                    android:textSize="20dp" />
                <EditText android:id="@+id/othercontact"
                    android:layout_width="fill_parent"
android:layout_height="wrap_content"
                    android:scrollbars="vertical" android:hint="其他联系
方式"
                    android:gravity="top" />
            </LinearLayout>
            <LinearLayout android:layout_width="fill_parent"
                android:layout_height="wrap_content"
android:orientation="vertical"
                android:layout_marginLeft="10dp"
android:layout_marginRight="10dp">
                <TextView android:layout_height="wrap_content"
                    android:layout_width="wrap_content" android:text="备
注："
                    android:textSize="20dp" />
                <EditText android:id="@+id/remark"
android:layout_width="fill_parent"
                    android:layout_height="wrap_content"
android:scrollbars="vertical"
                    android:hint="备注" android:gravity="top" />
            </LinearLayout>
        </LinearLayout>
</ScrollView>
<LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
       android:orientation="horizontal">
       <Button android:id="@+id/save"
android:layout_width="fill_parent"
            android:layout_height="wrap_content"
android:layout_weight="1"
            android:text="修改" />
       <Button android:id="@+id/delete"
android:layout_width="fill_parent"
            android:layout_height="wrap_content"
android:layout_weight="1"
            android:text="删除" />
```

```
        <Button android:id="@+id/btn_return"
android:layout_width="fill_parent"
            android:layout_height="wrap_content"
android:layout_weight="1"
            android:text="返回" />
    </LinearLayout>
</LinearLayout>
```

2.7 backup_progress.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
android:id="@+id/widget29"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical"
xmlns:android="http://schemas.android.com/apk/res/android"
>
<ProgressBar
android:id="@+id/pb_backup"
style="?android:attr/progressBarStyleHorizontal"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
>
</ProgressBar>
<Button
android:id="@+id/btn_backuup_ok"
android:layout_width="fill_parent"
android:gravity="center_horizontal"
android:layout_height="wrap_content"
android:text="OK"
android:visibility="gone"
>
</Button>
</LinearLayout>
```

2.8 enterfilename.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
android:id="@+id/widget29"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical"
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
>
<TextView
    android:id="@+id/tv_enter_file_name"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="请输入备份文件的文件名\n该文件该置于SD卡的根目录zpContact文
件夹下">
</TextView>
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/et_enter_file_name"
        >
</EditText>
</LinearLayout>
```

2.9 listitem.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
android:id="@+id/widget33"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
xmlns:android="http://schemas.android.com/apk/res/android"
>

    <ImageView
    android:id="@+id/user_image"
    android:layout_width="50px"
    android:layout_height="53px"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    >
    </ImageView>

    <LinearLayout
    android:id="@+id/linerlayout"
    android:layout_width="fill_parent"
    android:layout_height="53px"
    android:orientation="vertical"
    android:layout_alignParentTop="true"
    android:layout_toRightOf="@+id/user_image"
    android:layout_alignParentRight="true"
```

```
>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="25px"
        android:orientation="horizontal"
        >

        <TextView
        android:id="@+id/tv_showname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="17sp"
        android:gravity="top"
        android:text="姓名:"
        />

        <TextView
        android:id="@+id/tv_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="7"
        android:textSize="17sp"
        android:gravity="left"
        />

        </LinearLayout>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="25px"
        android:orientation="horizontal"
        >

        <TextView
        android:id="@+id/tv_showmobilephone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="17sp"
        android:gravity="left"
        android:text="手机:"
        />
    <TextView
        android:id="@+id/tv_mobilephone"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="7"
            android:textSize="20sp"
            android:gravity="left"
            />
            </LinearLayout>
    </LinearLayout>

<ImageView
    android:id="@+id/user_mark"
    android:layout_width="60px"
    android:layout_height="15px"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:visibility="gone"
    android:src="@drawable/checkmark"
    >
    </ImageView>
</RelativeLayout>
```

2.10 login.xml


```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
android:id="@+id/widget29"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
xmlns:android="http://schemas.android.com/apk/res/android"
>
<TextView
android:id="@+id/tv_account"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="请输入银行帐号"
```

```
android:textSize="16sp"
android:textStyle="bold"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
>
</TextView>
<EditText
android:id="@+id/et_account"
android:layout_width="300px"
android:layout_height="wrap_content"
android:textSize="18sp"
android:layout_below="@+id/tv_account"
android:layout_centerHorizontal="true"
>
</EditText>
<TextView
android:id="@+id/tv_password"
android:layout_width="100px"
android:layout_height="wrap_content"
android:text="密码"
android:textSize="16sp"
android:typeface="serif"
android:textStyle="bold"
android:gravity="center"
android:layout_below="@+id/et_account"
android:layout_alignLeft="@+id/tv_account"
>
</TextView>


<EditText
android:id="@+id/et_password"
android:layout_width="300px"
android:layout_height="wrap_content"
android:textSize="18sp"
android:password="true"
android:layout_below="@+id/tv_password"
android:layout_alignLeft="@+id/et_account"
>
</EditText>

<Button
android:id="@+id/btn_login_ok"
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:text="确定"
android:layout_below="@+id/et_password"
android:layout_toLeftOf="@+id/tv_password"
>
</Button>
<Button
android:id="@+id/btn_login_cancel"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="取消"
android:layout_alignTop="@+id/btn_login_ok"
android:layout_alignRight="@+id/tv_account"
>
</Button>
</RelativeLayout>
```

2.11 numchoose.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
android:id="@+id/widget28"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical"
xmlns:android="http://schemas.android.com/apk/res/android"
>
<TextView
android:id="@+id/tv_numchoose"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="请选择一个号码"
android:textSize="17sp"
android:typeface="serif"
android:gravity="center"
>
</TextView>
<ListView
android:id="@+id/num_list"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
>
</ListView>
</LinearLayout>
```

3. Activity 设计

3.1 AddNew.java

```
public class AddNew extends Activity implements ViewFactory {

    EditText et_name;
    EditText et_mobilePhone;
    EditText et_officePhone;
    EditText et_familyPhone;
    EditText et_position;
    EditText et_company;
    EditText et_address;
    EditText et_zipCode;
    EditText et_otherContact;
    EditText et_email;
    EditText et_remark;
    Button btn_save;
    Button btn_return;

    int privacy;//用于判断添加的用户是不是保密的
    ImageButton imageButton;//头像按钮
    View imageChooseView;//图像选择的视图
    AlertDialog imageChooseDialog;//头像选择对话框
    Gallery gallery;//头像的Gallery
    ImageSwitcher is;//头像的ImageSwitcher
    int currentImagePosition;//用于记录当前选中图像在图像数组中的位置
    int previousImagePosition;//用于记录上一次图片的位置
    boolean imageChanged;//判断头像有没有变化
    /**
     * 所有的图像图片
     */
    private  int[] images
            = new int[]{R.drawable.icon
        ,R.drawable.image1,R.drawable.image2,R.drawable.image3
        ,R.drawable.image4,R.drawable.image5,R.drawable.image6
        ,R.drawable.image7,R.drawable.image8,R.drawable.image9
        ,R.drawable.image10,R.drawable.image11,R.drawable.image12
        ,R.drawable.image13,R.drawable.image14,R.drawable.image15
        ,R.drawable.image16,R.drawable.image17,R.drawable.image18
        ,R.drawable.image19,R.drawable.image20,R.drawable.image21
        ,R.drawable.image22,R.drawable.image23,R.drawable.image24
        ,R.drawable.image25,R.drawable.image26,R.drawable.image27
```

```
        ,R.drawable.image28,R.drawable.image29,R.drawable.image30};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.addnew);
        Intent intent = getIntent();
        //先确定好保密类型
        if(intent.getExtras()!=null &&
intent.getExtras().getInt("privacy") == 1) {
            privacy = 1;
        } else {
            privacy = 0;
        }


        et_name = (EditText)findViewById(R.id.username);
        et_mobilePhone = (EditText)findViewById(R.id.mobilephone);
        et_officePhone = (EditText)findViewById(R.id.officephone);
        et_familyPhone = (EditText)findViewById(R.id.familyphone);
        et_position = (EditText)findViewById(R.id.position);
        et_company = (EditText)findViewById(R.id.company);
        et_address = (EditText)findViewById(R.id.address);
        et_zipCode = (EditText)findViewById(R.id.zipcode);
        et_otherContact = (EditText)findViewById(R.id.othercontact);
        et_email = (EditText)findViewById(R.id.email);
        et_remark = (EditText)findViewById(R.id.remark);

        btn_save = (Button)findViewById(R.id.save);
        btn_return = (Button)findViewById(R.id.btn_return);
        imageButton = (ImageButton)findViewById(R.id.image_button);

        /**
         * 响应点击事件
         */
        btn_save.setOnClickListener(new OnClickListener(){
            @Override
            public void onClick(View v) {
                //判断姓名是否为空
                String name = et_name.getText().toString();
                if(name.trim().equals("")) {
Toast.makeText(AddNew.this, "姓名不许为空", Toast.LENGTH_LONG).show();
                    return;
                }
```

```
                //从表单上获取数据
                User user = new User();
                user.username = name;
                user.address = et_address.getText().toString();
                user.company = et_company.getText().toString();
                user.email = et_email.getText().toString();
                user.familyPhone = et_familyPhone.getText().toString();
                user.mobilePhone = et_mobilePhone.getText().toString();
                user.officePhone = et_officePhone.getText().toString();
               user.otherContact = et_otherContact.getText().toString();
                user.position = et_position.getText().toString();
                user.remark = et_remark.getText().toString();
                user.zipCode = et_zipCode.getText().toString();
//判断头像是否改变，若改变，则用当前的位置，若没有改变，则用前一回的位置
                if(imageChanged) {
                    user.imageId =
images[currentImagePosition%images.length];
                } else {
                    user.imageId =
images[previousImagePosition%images.length];
                }
                user.privacy = privacy;
                //创建数据库帮助类
                DBHelper helper = new DBHelper(AddNew.this);
                //打开数据库
                helper.openDatabase();
                //把user存储到数据库里
                long result = helper.insert(user);

                //通过结果来判断是否插入成功，若为1，则表示插入数据失败
                if(result == -1 ) {
Toast.makeText(AddNew.this, "添加失败", Toast.LENGTH_LONG);
                }
                setTitle("用户添加成功!");
                //返回到上一个Activity，也就是Main.activity
                setResult(3);
                //销毁当前视图
                finish();
            }

        });

        btn_return.setOnClickListener(new OnClickListener(){
            @Override
```

```
            public void onClick(View v) {
                finish();
            }
        });

        imageButton.setOnClickListener(new OnClickListener(){
            @Override
            public void onClick(View v) {

                loadImage();//为gallery装载图片
                initImageChooseDialog();//初始化imageChooseDialog
                imageChooseDialog.show();
            }
        });

    }

    public void loadImage() {
        if(imageChooseView == null) {
            LayoutInflater li = LayoutInflater.from(AddNew.this);
            imageChooseView = li.inflate(R.layout.imageswitch, null);

            //通过渲染xml文件，得到一个视图（View），再拿到这个View里面的Gallery
            gallery =
(Gallery)imageChooseView.findViewById(R.id.gallery);
            //为Gallery装载图片
            gallery.setAdapter(new ImageAdapter(this));
            gallery.setSelection(images.length/2);
is = (ImageSwitcher)imageChooseView.findViewById(R.id.imageswitch);
            is.setFactory(this);
is.setInAnimation(AnimationUtils.loadAnimation(this,
android.R.anim.fade_in));
            //卸载图片的动画效果
is.setOutAnimation(AnimationUtils.loadAnimation(this,
android.R.anim.fade_out));
gallery.setOnItemSelectedListener(new OnItemSelectedListener(){
                @Override
                public void onItemSelected(AdapterView<?> arg0, View arg1,
                        int arg2, long arg3) {
                    //当前的头像位置为选中的位置
                    currentImagePosition = arg2;
                    //为ImageSwitcher设置图像
                    is.setImageResource(images[arg2 % images.length]);
```

```
            }
            @Override
            public void onNothingSelected(AdapterView<?> arg0) {

            }});
        }

    }

    /**
     * 自定义Gallery的适配器
     * @author Administrator
     *
     */
    class ImageAdapter extends BaseAdapter {

        private Context context;

        public ImageAdapter(Context context) {
            this.context = context;
        }

        @Override
        public int getCount() {
            return Integer.MAX_VALUE;
        }

        @Override
        public Object getItem(int position) {
            return position;
        }

        @Override
        public long getItemId(int position) {
            return position;
        }


        /**
         * gallery从这个方法中拿到image
         */
        @Override
        public View getView(int position, View convertView, ViewGroup
parent) {
```

```
            ImageView iv = new ImageView(context);
            iv.setImageResource(images[position%images.length]);
            iv.setAdjustViewBounds(true);
            iv.setLayoutParams(new Gallery.LayoutParams(80,80));
            iv.setPadding(15, 10, 15, 10);
            return iv;

        }

    }


    @Override
    public View makeView() {
        ImageView view = new ImageView(this);
        view.setBackgroundColor(0xff000000);
        view.setScaleType(ScaleType.FIT_CENTER);
        view.setLayoutParams(new ImageSwitcher.LayoutParams(90,90));
        return view;
    }

    public void initImageChooseDialog() {
        if(imageChooseDialog == null) {
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setTitle("请选择图像")
            .setView(imageChooseView).setPositiveButton("确定", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    imageChanged = true;
                    previousImagePosition = currentImagePosition;

    imageButton.setImageResource(images[currentImagePosition%images.l
ength]);
                }
            })
            .setNegativeButton("取消", new
DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface dialog, int which) {
                    currentImagePosition = previousImagePosition;

                }
```

```
            });
            imageChooseDialog = builder.create();
        }
    }

    /**
     * 当退出的时候，回收资源
     */
    @Override
    protected void onDestroy() {
        if(is != null) {
            is = null;
        }
        if(gallery != null) {
            gallery = null;
        }
        if(imageChooseDialog != null) {
            imageChooseDialog = null;
        }
        if(imageChooseView != null) {
            imageChooseView = null;
        }
        if(imageButton != null) {
            imageButton = null;
        }

        super.onDestroy();
    }
}
```

3.2 Main.java

```
public class Main extends Activity {
//显示所有数据的ListView
    ListView lv;

    ArrayList list;

    //拥有所有数据的Adapter
    SimpleAdapter adapter;
    //屏幕下方的工具栏
    GridView bottomMenuGrid;
    //主菜单的布局
```

```
    GridView mainMenuGrid;
    //主菜单的视图
    View mainMenuView;
    //登录的视图
    View loginView;

    //装搜索框的linearlayout,默认情况下visibility=gone
    LinearLayout searchLinearout;
    LinearLayout mainLinearLayout;
    //搜索框
    EditText et_search;
    EditText et_enter_file_name;

    //主菜单的对话框
    AlertDialog mainMenuDialog;
    //确认对话框
    AlertDialog confirmDialog;
    //进度条对话框
    AlertDialog progressDialog;
    //输入文件名的对话框
    AlertDialog enterFileNameDialog;
    //输入用户名密码的对话框
    AlertDialog loginDialog;
    //表示保密状态
    boolean privacy = false;
    //存储标记的数目
    int markedNum;
    //存储标记条目的_id号
    ArrayList<Integer> deleteId;
    // 菜单文字
    String[] main_menu_itemName = { "显示所有", "删除所有", "备份数据", "
还原数据", "更新", "后退"};
    //主菜单图片
    int[] main_menu_itemSource = {
                            R.drawable.showall,
                            R.drawable.menu_delete,
                            R.drawable.menu_backup,
                            R.drawable.menu_restore,
                            R.drawable.menu_fresh,
                            R.drawable.menu_return};

    String[] bottom_menu_itemName = { "增加", "查找", "删除", "菜单","退
出" };
    String fileName;
```

```
    int[] bottom_menu_itemSource = {
                                R.drawable.menu_new_user,
                                R.drawable.menu_search,
                                R.drawable.menu_delete,
                                R.drawable.controlbar_showtype_list,
                                R.drawable.menu_exit };


    /**
     * onCreate做的工作就是把listView显示出来
     * bottomMenuGrid，mainMenuGrid，searchLinearout都是到要用
     * 的时候再初始化，并且只初始化一次
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mainLinearLayout = (LinearLayout)findViewById(R.id.list_ll);
        DBHelper helper = new DBHelper(this);//获得所有用户的list
        helper.openDatabase(); //打开数据库，就打开这一次，因为Helper中的
SQLiteDatabase是静态的。
        list = helper.getAllUser(privacy);//拿到所有保密状态为privacy的用户
的list

        lv = (ListView)findViewById(R.id.lv_userlist); //创建ListView对
象
        if(list.size() == 0) {
Drawable nodata_bg = getResources().getDrawable(R.drawable.nodata_bg);
            mainLinearLayout.setBackgroundDrawable(nodata_bg);
            setTitle("没有查到任何数据");
        }
        //将数据与adapter集合起来
adapter = new SimpleAdapter(this,
                                    list,
                                    R.layout.listitem,
                                    new
String[]{"imageid","name","mobilephone"},
                                    new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});

lv.setAdapter(adapter);//将整合好的adapter交给listview，显示给用户看
```

```
        lv.setOnItemClickListener(new OnItemClickListener() {
        /**
         * 响应单击事件，单点击某一个选项的时候，跳转到用户详细信息页面
         */
            @Override
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
                    long arg3) {
                HashMap item = (HashMap)arg0.getItemAtPosition(arg2);
int _id = Integer.parseInt(String.valueOf(item.get("_id")));
Intent intent = new Intent(Main.this,UserDetail.class);
                User user = new User();
                user._id =
Integer.parseInt(String.valueOf(item.get("_id")));
                user.address = String.valueOf(item.get("address"));
                user.company = String.valueOf(item.get("company"));
                user.email = String.valueOf(item.get("email"));
                user.familyPhone =
String.valueOf(item.get("familyphone"));
user.mobilePhone = String.valueOf(item.get("mobilephone"));
user.officePhone = String.valueOf(item.get("officephone"));
user.otherContact = String.valueOf(item.get("othercontact"));
                user.position = String.valueOf(item.get("position"));
                user.remark = String.valueOf(item.get("remark"));
                user.username = String.valueOf(item.get("name"));
                user.zipCode = String.valueOf(item.get("zipcode"));
user.imageId = Integer.parseInt(String.valueOf(item.get("imageid")));

                intent.putExtra("user", user);

if(searchLinearout != null &&
searchLinearout.getVisibility()==View.VISIBLE) {
                    searchLinearout.setVisibility(View.GONE);
                }

                /*将arg2作为请求码传过去  用于标识修改项的位置*/
                startActivityForResult(intent, arg2);
            }
        });

        lv.setCacheColorHint(Color.TRANSPARENT); //设置ListView的背景为透
明
        lv.setOnItemLongClickListener(new OnItemLongClickListener(){
```

```
            @Override
            public boolean onItemLongClick(AdapterView<?> arg0, View
arg1,
                    int arg2, long arg3) {
                if(deleteId == null) {
                    deleteId = new ArrayList<Integer>();
                }
                HashMap item = (HashMap)arg0.getItemAtPosition(arg2);
Integer _id = Integer.parseInt(String.valueOf(item.get("_id")));

                RelativeLayout r = (RelativeLayout)arg1;
                ImageView markedView = (ImageView)r.getChildAt(2);
                if(markedView.getVisibility() == View.VISIBLE) {
                    markedView.setVisibility(View.GONE);
                    deleteId.remove(_id);
                } else {
                    markedView.setVisibility(View.VISIBLE);
                    deleteId.add(_id);
                }
                return true;
            }


        });
        //为list添加item选择器
        Drawable bgDrawable =
getResources().getDrawable(R.drawable.list_bg);
        lv.setSelector(bgDrawable);

    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        //清除deleteId的内容
        if(deleteId != null) {
            deleteId.clear();
        }
        //当resultCode==3时代表添加了一个用户返回,当resultCode==4的时候代表修
改了用户，或者删除了用户，其他条件代表数据没有变化
        if(resultCode == 3 || resultCode == 4) {
            DBHelper helper = new DBHelper(this);
            list = helper.getAllUser(privacy);
```

```
            adapter =
            new SimpleAdapter(
                                this,
                                list,
                                R.layout.listitem,
                                new
String[]{"imageid","name","mobilephone"},
                                new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});
            if(list.size() > 0){
            mainLinearLayout.setBackgroundDrawable(null);
            }
        }

        lv.setAdapter(adapter); //将整合好的adapter交给listview，显示给用户
看
        /**
         * resultCode只有3、4、5
         * 当等于4或者5的时候,代表由UserDetail转过来的。在转想UserDetail的时候,
requestCode的值设置的是选中项的位置
         */
        if(resultCode == 3) {
            lv.setSelection(list.size());
        } else {
            lv.setSelection(requestCode);
        }


    }

    /**
     * 响应点击Menu按钮时的事件，用于设置底部菜单是否可见
     */
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if(keyCode == KeyEvent.KEYCODE_MENU) {
            loadBottomMenu();
            if(bottomMenuGrid.getVisibility() == View.VISIBLE) {
                if(searchLinearout != null &&
searchLinearout.getVisibility() == View.VISIBLE) {
                    searchLinearout.setVisibility(View.GONE);
                }
                bottomMenuGrid.setVisibility(View.GONE);
            } else {
```

```
                bottomMenuGrid.setVisibility(View.VISIBLE);
            }
        }
        return super.onKeyDown(keyCode, event);
    }


    private void loadBottomMenu() {

        if(bottomMenuGrid == null) {
            bottomMenuGrid = (GridView)
findViewById(R.id.gv_buttom_menu);

    bottomMenuGrid.setBackgroundResource(R.drawable.channelgallery_bg)
;// 设置背景
            bottomMenuGrid.setNumColumns(5);// 设置每行列数
            bottomMenuGrid.setGravity(Gravity.CENTER);// 位置居中
            bottomMenuGrid.setVerticalSpacing(10);// 垂直间隔
            bottomMenuGrid.setHorizontalSpacing(10);// 水平间隔

    bottomMenuGrid.setAdapter(getMenuAdapter(bottom_menu_itemName,
                    bottom_menu_itemSource));// 设置菜单Adapter
            /** 监听底部菜单选项 **/
            bottomMenuGrid.setOnItemClickListener(new
OnItemClickListener() {
                public void onItemClick(AdapterView<?> arg0, View arg1,
int arg2,
                        long arg3) {

                    switch (arg2) {
                        case 0: {
if(searchLinearout != null &&
searchLinearout.getVisibility()==View.VISIBLE) {
    searchLinearout.setVisibility(View.GONE);
                            }
if(bottomMenuGrid.getVisibility() == View.VISIBLE) {
    bottomMenuGrid.setVisibility(View.GONE);
                            }

Intent intent = new Intent(Main.this,AddNew.class);
startActivityForResult(intent, 3);
                            break;
                        }
```

```
                    case 1:
                        loadSearchLinearout();

    if(searchLinearout.getVisibility()==View.VISIBLE) {
                            searchLinearout.setVisibility(View.GONE);
                        } else {

                       searchLinearout.setVisibility(View.VISIBLE);
                            et_search.requestFocus();
                            et_search.selectAll();
                        }
                        break;
                    case 2:
                        if(searchLinearout != null &&
searchLinearout.getVisibility()==View.VISIBLE) {
                            searchLinearout.setVisibility(View.GONE);
                        }
                        if(deleteId == null || deleteId.size() == 0) {
Toast.makeText(Main.this, "    没有标记任何记录\n长按一条记录即可标记",
Toast.LENGTH_LONG).show();
                        } else {
new AlertDialog.Builder(Main.this).setTitle("确定要删除标记的
"+deleteId.size()+"条记录吗?").setPositiveButton("确定", new
DialogInterface.OnClickListener() {
                                @Override
public void onClick(DialogInterface dialog, int which) {
                        DBHelper helper = new DBHelper(Main.this);
                                helper.deleteMarked(deleteId);
                                //重置视图
                                list = helper.getAllUser(privacy);
                                adapter =
                                new SimpleAdapter(
                                                Main.this,
                                                list,
                                                R.layout.listitem,
                                                new
String[]{"imageid","name","mobilephone"},
                                                new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});
                                lv.setAdapter(adapter);
                                deleteId.clear();
                            }
                        })
                        .setNegativeButton("取消", null)
```

```
                        .create()
                        .show()    ;
                    }

                    break;
                case 3:
                    if(searchLinearout != null &&
searchLinearout.getVisibility()==View.VISIBLE) {
                        searchLinearout.setVisibility(View.GONE);
                    }
                    loadMainMenuDialog();
                    mainMenuDialog.show();

                    break;
                case 4:
                    finish();
                    break;
                }
            }
        });
    }

}

    private void loadMainMenuDialog() {
        if(mainMenuDialog == null) {
            LayoutInflater li = LayoutInflater.from(this);
            mainMenuView = li.inflate(R.layout.main_menu_grid, null);
             //根据主菜单视图，创建主菜单对话框
            mainMenuDialog = new
AlertDialog.Builder(this).setView(mainMenuView).create();
            //根据主菜单视图，拿到视图文件中的GridView，然后再往里面放Adapter
            mainMenuGrid =
(GridView)mainMenuView.findViewById(R.id.gridview);
            SimpleAdapter menuAdapter =
getMenuAdapter(main_menu_itemName, main_menu_itemSource);
            mainMenuGrid.setAdapter(menuAdapter);
            //响应点击事件
            mainMenuGrid.setOnItemClickListener(new
OnItemClickListener(){
                @Override
                public void onItemClick(AdapterView<?> arg0, View arg1,
int arg2,
                        long arg3) {
```

```
                switch(arg2){
                    case 0:{
                         DBHelper helper = new DBHelper(Main.this);
                         list = helper.getAllUser(privacy);
                         adapter = new SimpleAdapter(
                              Main.this,
                              list,
                              R.layout.listitem,
                              new
String[]{"imageid","name","mobilephone"},
                              new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});

                         lv.setAdapter(adapter);//显示所有数据
                         mainMenuDialog.dismiss();
                        break;
                    }
                    case 1:{
                        AlertDialog.Builder builder  = new
AlertDialog.Builder(Main.this);
                        confirmDialog = builder.create();
                        builder.setTitle("是否删除所有！?");
                        builder.setPositiveButton("确定", new
DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface
dialog, int which) {
                                    DBHelper helper = new
DBHelper(Main.this);
                                   helper.deleteAll(0);
                                    list = helper.getAllUser(privacy);
                                    adapter = new SimpleAdapter(
                                         Main.this,
                                         list,
                                         R.layout.listitem,
                                         new
String[]{"imageid","name","mobilephone"},
                                         new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});

                                    lv.setAdapter(adapter);//显示所有数
据
                                   mainMenuDialog.dismiss();
                                }
```

```
                            });
                        builder.setNegativeButton("取消", new
DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface
dialog, int which) {
                                    confirmDialog.dismiss();
                                }
                            });
                        builder.create().show();
                        break;
                    }
                    case 2:{
                        mainMenuDialog.dismiss();
                        new AlertDialog.Builder(Main.this)
                        .setTitle("是否需要备份记录到SD卡？")
                        .setPositiveButton("确定", new
DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog,
int which) {
                            LayoutInflater li =
LayoutInflater.from(Main.this);
                            View backup_view =
li.inflate(R.layout.backup_progress, null);
                            progressDialog =  new
AlertDialog.Builder(Main.this)
                            .setTitle("备份正在进行中...")
                            .setView(backup_view)
                            .create();
                            progressDialog.show();
                            DBHelper helper = new DBHelper(Main.this);
                            helper.backupData(privacy);
                            ProgressBar bar = (ProgressBar)
backup_view.findViewById(R.id.pb_backup);
                            Button btn_backup_ok =
(Button)backup_view.findViewById(R.id.btn_backuup_ok);
                            bar.setMax(list.size());
                            for(int i=0;i<=list.size();i++) {
                                bar.setProgress(i);
                            }
                            progressDialog.setTitle("备份完成！一共 "+
list.size() + " 条记录");
```

```
    btn_backup_ok.setVisibility(View.VISIBLE);
                            btn_backup_ok.setOnClickListener(new
OnClickListener() {

                                @Override
                                public void onClick(View v) {
                                    progressDialog.dismiss();
                                    mainMenuDialog.dismiss();
                                }

                            });
                            }

                        })
                        .setNegativeButton("取消", null)
                        .create()
                        .show()    ;
                        break;
                    }
                    case 3:{
                        LayoutInflater li =
LayoutInflater.from(Main.this);
                        View enterFileNameView =
li.inflate(R.layout.enterfilename, null);
                        enterFileNameDialog =    new
AlertDialog.Builder(Main.this)

    .setView(enterFileNameView).setNegativeButton("取消", null)
                        .setPositiveButton("确定", new
DialogInterface.OnClickListener() {

                            @Override
                            public void onClick(DialogInterface dialog,
int which) {
                                DBHelper helper = new
DBHelper(Main.this);
                                fileName =
et_enter_file_name.getText().toString();
                                if(helper.findFile(fileName)){
                                    new
AlertDialog.Builder(Main.this).setTitle("请选择方式")
                                        .setPositiveButton("覆盖", new
DialogInterface.OnClickListener() {
```

```
                                        @Override
                                        public void
onClick(DialogInterface dialog, int which) {
                                            DBHelper helper = new
DBHelper(Main.this);
                                            helper.deleteAll(0);

    helper.restoreData(fileName);
                                            list =
helper.getAllUser(privacy);
                                            adapter = new
SimpleAdapter(Main.this,
                                                    list,
                                                    R.layout.listitem,
                                                    new
String[]{"imageid","name","mobilephone"},
                                                    new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});
                                            LayoutInflater li =
LayoutInflater.from(Main.this);
                                            View backup_view =
li.inflate(R.layout.backup_progress, null);
                                            progressDialog =  new
AlertDialog.Builder(Main.this)
                                            .setTitle("正在还原数据...")
                                            .setView(backup_view)
                                            .create();
                                            progressDialog.show();
                                            ProgressBar bar =
(ProgressBar) backup_view.findViewById(R.id.pb_backup);

                                            Button btn_backup_ok =
(Button)backup_view.findViewById(R.id.btn_backuup_ok);
                                            bar.setMax(list.size());
                                            for(int
i=0;i<=list.size();i++) {
                                                bar.setProgress(i);
                                            }
                                            progressDialog.setTitle("还
原完成！一共还原了 "+ list.size() + " 条记录");

    btn_backup_ok.setVisibility(View.VISIBLE);

    btn_backup_ok.setOnClickListener(new OnClickListener() {
```

```
                                        @Override
                                        public void onClick(View
v) {
    progressDialog.dismiss();
    mainMenuDialog.dismiss();
                                            if(list.size() != 0)
{
    mainLinearLayout.setBackgroundDrawable(null);
                                            }
    lv.setAdapter(adapter);
                                        }
                                    });
                                }
                            })
                            .setNegativeButton("添加", new
DialogInterface.OnClickListener() {
                                @Override
                                public void
onClick(DialogInterface dialog, int which) {
                                    DBHelper helper = new
DBHelper(Main.this);
                                    int preNum = list.size();
    helper.restoreData(fileName);
                                    list =
helper.getAllUser(privacy);
                                    adapter = new
SimpleAdapter(Main.this,
                                            list,
                                            R.layout.listitem,
                                            new
String[]{"imageid","name","mobilephone"},
                                            new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});
                                    LayoutInflater li =
LayoutInflater.from(Main.this);
```

```
                                        View backup_view =
li.inflate(R.layout.backup_progress, null);
                                        progressDialog =  new
AlertDialog.Builder(Main.this)
                                        .setTitle("正在还原数据...")
                                        .setView(backup_view)
                                        .create();
                                        progressDialog.show();
                                        ProgressBar bar =
(ProgressBar) backup_view.findViewById(R.id.pb_backup);

                                        Button btn_backup_ok =
(Button)backup_view.findViewById(R.id.btn_backuup_ok);
                                        bar.setMax(list.size());
                                        for(int
i=0;i<=list.size();i++) {
                                            bar.setProgress(i);
                                        }
                                        progressDialog.setTitle("还
原完成！一共还原了 "+ (list.size()-preNum) + " 条记录");

    btn_backup_ok.setVisibility(View.VISIBLE);

    btn_backup_ok.setOnClickListener(new OnClickListener() {

                                            @Override
                                            public void onClick(View
v) {

    progressDialog.dismiss();

    mainMenuDialog.dismiss();

    lv.setAdapter(adapter);
                                            }

                                        });
                                    }
                                })
                                .setNeutralButton("取消", new
DialogInterface.OnClickListener() {

                                    @Override
                                    public void
```

```
onClick(DialogInterface dialog, int which) {

                                        }
                                    }).create().show();

                                } else {

    Toast.makeText(enterFileNameDialog.getContext(), "找不到备份文件",
Toast.LENGTH_LONG).show();
                                }
                            }
                        })
                        .create();
                        et_enter_file_name =
(EditText)enterFileNameView.findViewById(R.id.et_enter_file_name);
                        et_enter_file_name.setText("comm_data");
                        et_enter_file_name.requestFocus();
                        et_enter_file_name.selectAll();
                        enterFileNameDialog.show();
                        adapter = new SimpleAdapter(
                              Main.this,
                              list,
                              R.layout.listitem,
                              new
String[]{"imageid","name","mobilephone"},
                              new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});

                         lv.setAdapter(adapter);//显示所有数据
                        mainMenuDialog.dismiss();
                        break;
                    }
                    case 4:{
                        mainMenuDialog.dismiss();
                        new AlertDialog.Builder(Main.this)
                        .setTitle("更新操作将需要支付20元的费用！是否继续?
")
                        .setPositiveButton("确定", new
DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog,
int which) {
                                //新建一个activity出来
                                LayoutInflater li =
```

```
LayoutInflater.from(Main.this);
                                loginView = li.inflate(R.layout.login,
null);

                                Button btn_login_ok =
(Button)loginView.findViewById(R.id.btn_login_ok);
                                Button btn_login_cancel =
(Button)loginView.findViewById(R.id.btn_login_cancel);
                                final EditText et_account =
(EditText)loginView.findViewById(R.id.et_account);
                                final EditText et_password =
(EditText)loginView.findViewById(R.id.et_password);
                                btn_login_ok.setOnClickListener(new
OnClickListener(){

                                    @Override
                                    public void onClick(View v) {

    if(et_account.getText().toString().equals("admin") &&
et_password.getText().toString().equals("admin")) {
                                        et_account.setText("");
                                        et_password.setText("");
                                        loginDialog.dismiss();
                                        Intent intent = new
Intent(Main.this,MainPrivacy.class);
                                        startActivity(intent);

                                    } else {
                                        Toast.makeText(Main.this, "
失败", Toast.LENGTH_LONG).show();
                                    }
                                }

                                });

    btn_login_cancel.setOnClickListener(new OnClickListener(){

                                    @Override
                                    public void onClick(View v) {
                                        loginDialog.dismiss();
                                    }

                                });
```

```
                                if(loginDialog == null) {
                                    loginDialog = new
AlertDialog.Builder(Main.this).setView(loginView).create();
                                }
                                loginDialog.show();


                            }
                        })
                        .setNegativeButton("取消", null)
                        .create()
                        .show()    ;
                        break;
                    }
                    case 5:{
                        mainMenuDialog.dismiss();
                        break;
                    }
                }

            }});
        }

    }

    private void loadSearchLinearout() {

        if(searchLinearout == null) {
            searchLinearout = (LinearLayout)
findViewById(R.id.ll_search);
            et_search = (EditText)findViewById(R.id.et_search);
            et_search.setOnKeyListener(new OnKeyListener(){
                @Override
                public boolean onKey(View arg0, int arg1, KeyEvent arg2)
{
                    String condition = et_search.getText().toString();
                    if(condition.equals("")) {
                        lv.setAdapter(adapter);
                    }
                    DBHelper helper = new DBHelper(Main.this);
                    list = helper.getUsers(condition,privacy);
                    SimpleAdapter searchAdapter =
                        new SimpleAdapter(
                                Main.this,
```

```
                                list,
                                R.layout.listitem,
                                new
String[]{"imageid","name","mobilephone"},
                                new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});
                    lv.setAdapter(searchAdapter);  //将整合好的adapter交给
listview，显示给用户看
                    if(list.size() == 0) {
                        Drawable nodata_bg =
getResources().getDrawable(R.drawable.nodata_bg);

    mainLinearLayout.setBackgroundDrawable(nodata_bg);
                        setTitle("没有查到任何数据");
                    } else {
                        setTitle( "共查到 " + list.size()+" 条记录");

                        mainLinearLayout.setBackgroundDrawable(null);
                    }
                    return false;
                }});

        }

    }

    private SimpleAdapter getMenuAdapter(String[] menuNameArray,
            int[] imageResourceArray) {
        ArrayList<HashMap<String, Object>> data = new
ArrayList<HashMap<String, Object>>();
        for (int i = 0; i < menuNameArray.length; i++) {
            HashMap<String, Object> map = new HashMap<String, Object>();
            map.put("itemImage", imageResourceArray[i]);
            map.put("itemText", menuNameArray[i]);
            data.add(map);
        }
        SimpleAdapter simperAdapter =
                new SimpleAdapter(
                                this,
                                data,
                                R.layout.item_menu,
                                new String[] { "itemImage", "itemText" },
                                new int[] { R.id.item_image,
R.id.item_text });
```

```
    return simperAdapter;
}

/**
 * 当退出的时候，回收资源
 */
@Override
protected void onDestroy() {
    if(confirmDialog != null) {
        confirmDialog = null;
    }
    if(mainMenuDialog != null) {
        mainMenuDialog = null;
    }
    if(searchLinearout != null) {
        searchLinearout = null;
    }
    if(mainMenuView != null) {
        mainMenuView = null;
    }
    if(mainMenuGrid != null) {
        mainMenuGrid = null;
    }
    if(bottomMenuGrid != null) {
        bottomMenuGrid = null;
    }
    if(adapter != null) {
        adapter = null;
    }
    if(list != null) {
        list = null;
    }
    if(lv != null) {
        lv = null;
    }
    if(DBHelper.dbInstance != null) {
        DBHelper.dbInstance.close();
        DBHelper.dbInstance = null;
    }

    System.out.println("destory!!!");
    super.onDestroy();
}
```

```
}
```

3.3 MainPrivacy.java

```
public class MainPrivacy extends Activity {



    //显示所有数据的ListView
    ListView lv;

    ArrayList list;

    //拥有所有数据的Adapter
    SimpleAdapter adapter;
    //屏幕下方的工具栏
    GridView bottomMenuGrid;
    //主菜单的布局
    GridView mainMenuGrid;
    //主菜单的视图
    View mainMenuView;

    View loginView;

    //装搜索框的linearlayout,默认情况下visibility=gone
    LinearLayout searchLinearout;
    LinearLayout mainLinearLayout;
    //搜索框
    EditText et_search;
    EditText et_enter_file_name;

    //主菜单的对话框
    AlertDialog mainMenuDialog;
    AlertDialog confirmDialog;
    AlertDialog progressDialog;
    AlertDialog enterFileNameDialog;
    AlertDialog loginDialog;

    boolean privacy = true;

    int i;
    ArrayList<Integer> deleteId;
    /** 菜单文字 **/
    String[] main_menu_itemName = { "显示所有", "删除所有", "备份数据", "
```

```
还原数据", "退出秘密仓库", "后退"};
    //主菜单图片
    int[] main_menu_itemSource = {
                            R.drawable.showall,
                            R.drawable.menu_delete,
                            R.drawable.menu_backup,
                            R.drawable.menu_restore,
                            R.drawable.menu_quit,
                            R.drawable.menu_return};

    String[] bottom_menu_itemName = { "增加", "查找", "删除", "菜单","退
出" };
    String fileName;
    int[] bottom_menu_itemSource = {
                             R.drawable.menu_new_user,
                             R.drawable.menu_search,
                             R.drawable.menu_delete,
                             R.drawable.controlbar_showtype_list,
                             R.drawable.menu_exit };


    /**
     * onCreate做的工作就是把listView显示出来
     * bottomMenuGrid，mainMenuGrid，searchLinearout都是到要用
     * 的时候再初始化，并且只初始化一次
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        setTitle("秘密仓库");
        Toast.makeText(this, "成功进入秘密仓库",
Toast.LENGTH_LONG).show();
        mainLinearLayout = (LinearLayout)findViewById(R.id.list_ll);
        DBHelper helper = new DBHelper(this);//获得所有用户的list
        helper.openDatabase(); //打开数据库，就打开这一次，因为Helper中的
SQLiteDatabase是静态的。
        list = helper.getAllUser(privacy);//拿到所有用户的list

        lv = (ListView)findViewById(R.id.lv_userlist); //创建ListView对
象
        if(list.size() == 0) {
            Drawable nodata_bg =
getResources().getDrawable(R.drawable.nodata_bg);
```

```
            mainLinearLayout.setBackgroundDrawable(nodata_bg);
            setTitle("没有查到任何数据");
        }
        //将数据与adapter集合起来
        adapter = new SimpleAdapter(this,
                                    list,
                                    R.layout.listitem,
                                    new
String[]{"imageid","name","mobilephone"},
                                    new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});

        lv.setAdapter(adapter);//将整合好的adapter交给listview,显示给用户看



        lv.setOnItemClickListener(new OnItemClickListener() {
        /**
         * 响应单击事件，单点击某一个选项的时候，跳转到用户详细信息页面
         */
            @Override
            public void onItemClick(AdapterView<?> arg0, View arg1, int
arg2,
                    long arg3) {
                HashMap item = (HashMap)arg0.getItemAtPosition(arg2);
                int _id =
Integer.parseInt(String.valueOf(item.get("_id")));

                Intent intent = new
Intent(MainPrivacy.this,UserDetail.class);
                User user = new User();
                user._id =
Integer.parseInt(String.valueOf(item.get("_id")));
                user.address = String.valueOf(item.get("address"));
                user.company = String.valueOf(item.get("company"));
                user.email = String.valueOf(item.get("email"));
                user.familyPhone =
String.valueOf(item.get("familyphone"));
                user.mobilePhone =
String.valueOf(item.get("mobilephone"));
                user.officePhone =
String.valueOf(item.get("officephone"));
                user.otherContact =
String.valueOf(item.get("othercontact"));
```

```
                user.position = String.valueOf(item.get("position"));
                user.remark = String.valueOf(item.get("remark"));
                user.username = String.valueOf(item.get("name"));
                user.zipCode = String.valueOf(item.get("zipcode"));
                user.imageId =
Integer.parseInt(String.valueOf(item.get("imageid")));

                intent.putExtra("user", user);

                if(searchLinearout != null &&
searchLinearout.getVisibility()==View.VISIBLE) {
                    searchLinearout.setVisibility(View.GONE);
                }

                /*将arg2作为请求码传过去  用于标识修改项的位置*/
                startActivityForResult(intent, arg2);
            }
        });

        lv.setCacheColorHint(Color.TRANSPARENT); //设置ListView的背景为透
明
        lv.setOnItemLongClickListener(new OnItemLongClickListener(){

            @Override
            public boolean onItemLongClick(AdapterView<?> arg0, View
arg1,
                    int arg2, long arg3) {
                if(deleteId == null) {
                    deleteId = new ArrayList<Integer>();
                }
                HashMap item = (HashMap)arg0.getItemAtPosition(arg2);
                Integer _id =
Integer.parseInt(String.valueOf(item.get("_id")));

                RelativeLayout r = (RelativeLayout)arg1;
                ImageView iv = (ImageView)r.getChildAt(2);
                iv.setEnabled(false);
                if(iv.getVisibility() == View.VISIBLE) {
                    iv.setVisibility(View.GONE);
                    deleteId.remove(_id);
                } else {
                    iv.setVisibility(View.VISIBLE);
                    deleteId.add(_id);
                }
```

```
                return true;
            }


        });

        //为list添加item选择器
        Drawable bgDrawableSelector =
getResources().getDrawable(R.drawable.list_bg);
        lv.setSelector(bgDrawableSelector);

    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        //当resultCode==3时代表添加了一个用户返回,当resultCode==4的时候代表修
改了用户，或者删除了用户，其他条件代表数据没有变化
        if(deleteId != null) {
            deleteId.clear();
        }
        if(resultCode == 3 || resultCode == 4) {

            DBHelper helper = new DBHelper(this);
            list = helper.getAllUser(privacy);
            adapter =
            new SimpleAdapter(
                            this,
                            list,
                            R.layout.listitem,
                            new
String[]{"imageid","name","mobilephone"},
                            new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});
        }

        if(list.size()>0) {
            mainLinearLayout.setBackgroundDrawable(null);
        }

        lv.setAdapter(adapter); //将整合好的adapter交给listview，显示给用户
看
        /**
```

```
         * resultCode只有3、4、5
         * 当等于4或者5的时候,代表由UserDetail转过来的。在转想UserDetail的时候,
requestCode的值设置的是选中项的位置
         */
        if(resultCode == 3) {
            lv.setSelection(list.size());
        } else {
            lv.setSelection(requestCode);
        }


    }

    /**
     * 响应点击Menu按钮时的事件，用于设置底部菜单是否可见
     */
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if(keyCode == KeyEvent.KEYCODE_MENU) {
            loadBottomMenu();
            if(bottomMenuGrid.getVisibility() == View.VISIBLE) {
                if(searchLinearout != null &&
searchLinearout.getVisibility() == View.VISIBLE) {
                    searchLinearout.setVisibility(View.GONE);
                }
                bottomMenuGrid.setVisibility(View.GONE);
            } else {
                bottomMenuGrid.setVisibility(View.VISIBLE);
            }
        }
        return super.onKeyDown(keyCode, event);
    }


    private void loadBottomMenu() {

        if(bottomMenuGrid == null) {
            bottomMenuGrid = (GridView)
findViewById(R.id.gv_buttom_menu);

    bottomMenuGrid.setBackgroundResource(R.drawable.channelgallery_bg)
;// 设置背景
            bottomMenuGrid.setNumColumns(5);// 设置每行列数
            bottomMenuGrid.setGravity(Gravity.CENTER);// 位置居中
```

```
            bottomMenuGrid.setVerticalSpacing(10);// 垂直间隔
            bottomMenuGrid.setHorizontalSpacing(10);// 水平间隔

    bottomMenuGrid.setAdapter(getMenuAdapter(bottom_menu_itemName,
                    bottom_menu_itemSource));// 设置菜单Adapter
            /** 监听底部菜单选项 **/
            bottomMenuGrid.setOnItemClickListener(new
OnItemClickListener() {
                public void onItemClick(AdapterView<?> arg0, View arg1,
int arg2,
                        long arg3) {

                    switch (arg2) {
                        case 0: {
                            if(searchLinearout != null &&
searchLinearout.getVisibility()==View.VISIBLE) {
                                searchLinearout.setVisibility(View.GONE);
                            }
                            if(bottomMenuGrid.getVisibility() ==
View.VISIBLE) {
                                bottomMenuGrid.setVisibility(View.GONE);
                            }

                            Intent intent = new
Intent(MainPrivacy.this,AddNew.class);
                            intent.putExtra("privacy", 1);
                            startActivityForResult(intent, 3);
                            break;
                        }

                        case 1:
                            loadSearchLinearout();

    if(searchLinearout.getVisibility()==View.VISIBLE) {
                                searchLinearout.setVisibility(View.GONE);
                            } else {

    searchLinearout.setVisibility(View.VISIBLE);
                                et_search.requestFocus();
                                et_search.selectAll();
                            }
                            break;
                        case 2:
                            if(searchLinearout != null &&
```

```
searchLinearout.getVisibility()==View.VISIBLE) {
                            searchLinearout.setVisibility(View.GONE);
                        }
                        if(deleteId == null || deleteId.size() == 0) {
                            Toast.makeText(MainPrivacy.this, "    没有
标记任何记录\n长按一条记录即可标记", Toast.LENGTH_LONG).show();
                        } else {
                            new AlertDialog.Builder(MainPrivacy.this)
                            .setTitle("确定要删除标记的
"+deleteId.size()+"条记录吗?")
                            .setPositiveButton("确定", new
DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface
dialog, int which) {
                                    DBHelper helper = new
DBHelper(MainPrivacy.this);
                                    helper.deleteMarked(deleteId);
                                    //重置视图
                                    list = helper.getAllUser(privacy);
                                    adapter =
                                    new SimpleAdapter(
                                                    MainPrivacy.this,
                                                    list,
                                                    R.layout.listitem,
                                                    new
String[]{"imageid","name","mobilephone"},
                                                    new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});
                                        lv.setAdapter(adapter);
                                        deleteId.clear();
                                    }
                                })
                                .setNegativeButton("取消", null)
                                .create()
                                .show()    ;
                        }

                        break;
                    case 3:
                        if(searchLinearout != null &&
searchLinearout.getVisibility()==View.VISIBLE) {
                            searchLinearout.setVisibility(View.GONE);
                        }
```

```
                        loadMainMenuDialog();
                        mainMenuDialog.show();

                        break;
                    case 4:
                        finish();
                        break;
                }
            }
        });
    }

}

    private void loadMainMenuDialog() {
        if(mainMenuDialog == null) {
            LayoutInflater li = LayoutInflater.from(this);
            mainMenuView = li.inflate(R.layout.main_menu_grid, null);
             //根据主菜单视图，创建主菜单对话框
            mainMenuDialog = new
AlertDialog.Builder(this).setView(mainMenuView).create();
            //根据主菜单视图，拿到视图文件中的GridView，然后再往里面放Adapter
            mainMenuGrid =
(GridView)mainMenuView.findViewById(R.id.gridview);
            SimpleAdapter menuAdapter =
getMenuAdapter(main_menu_itemName, main_menu_itemSource);
            mainMenuGrid.setAdapter(menuAdapter);
            //响应点击事件
            mainMenuGrid.setOnItemClickListener(new
OnItemClickListener(){
                @Override
                public void onItemClick(AdapterView<?> arg0, View arg1,
int arg2,
                        long arg3) {
                    switch(arg2){
                        case 0:{
                             DBHelper helper = new
DBHelper(MainPrivacy.this);
                            list = helper.getAllUser(privacy);
                            adapter = new SimpleAdapter(
                                 MainPrivacy.this,
                                 list,
                                 R.layout.listitem,
                                 new
```

```
String[]{"imageid","name","mobilephone"},
                                new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});

                                lv.setAdapter(adapter);//显示所有数据
                                mainMenuDialog.dismiss();
                               break;
                           }
                           case 1:{
                                mainMenuDialog.dismiss();
                               AlertDialog.Builder builder  = new
AlertDialog.Builder(MainPrivacy.this);
                               confirmDialog = builder.create();
                               builder.setTitle("是否删除所有！?");
                               builder.setPositiveButton("确定", new
DialogInterface.OnClickListener() {
                                    @Override
                                    public void onClick(DialogInterface
dialog, int which) {
                                        DBHelper helper = new
DBHelper(MainPrivacy.this);
                                       helper.deleteAll(1);
                                        list = helper.getAllUser(privacy);
                                        adapter = new SimpleAdapter(
                                             MainPrivacy.this,
                                             list,
                                             R.layout.listitem,
                                             new
String[]{"imageid","name","mobilephone"},
                                             new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});

                                         lv.setAdapter(adapter);//显示所有数
据

                                    }
                                });
                               builder.setNegativeButton("取消", new
DialogInterface.OnClickListener() {
                                    @Override
                                    public void onClick(DialogInterface
dialog, int which) {
                                        confirmDialog.dismiss();
                                    }
```

```
                        });
                    builder.create().show();
                    break;
                }
                case 2:{
                    mainMenuDialog.dismiss();
                    new AlertDialog.Builder(MainPrivacy.this)
                    .setTitle("是否需要备份记录到SD卡？")
                    .setPositiveButton("确定", new
DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog,
int which) {
                        LayoutInflater li =
LayoutInflater.from(MainPrivacy.this);
                        View backup_view =
li.inflate(R.layout.backup_progress, null);
                        progressDialog =  new
AlertDialog.Builder(MainPrivacy.this)
                        .setTitle("备份正在进行中...")
                        .setView(backup_view)
                        .create();
                        progressDialog.show();
                        DBHelper helper = new
DBHelper(MainPrivacy.this);
                        helper.backupData(privacy);
                        ProgressBar bar = (ProgressBar)
backup_view.findViewById(R.id.pb_backup);
                        Button btn_backup_ok =
(Button)backup_view.findViewById(R.id.btn_backuup_ok);
                        bar.setMax(list.size());
                        for(int i=0;i<=list.size();i++) {
                            bar.setProgress(i);
                        }
                        progressDialog.setTitle("备份完成！一共 "+
list.size() + " 条记录");

    btn_backup_ok.setVisibility(View.VISIBLE);
                        btn_backup_ok.setOnClickListener(new
OnClickListener() {

                            @Override
                            public void onClick(View v) {
                                progressDialog.dismiss();
```

```
                                mainMenuDialog.dismiss();

                            }

                        });
                        }

                    })
                    .setNegativeButton("取消", null)
                    .create()
                    .show()    ;

                    break;
                }
                case 3:{
                    LayoutInflater li =
LayoutInflater.from(MainPrivacy.this);

                    View enterFileNameView =
li.inflate(R.layout.enterfilename, null);
                    enterFileNameDialog =    new
AlertDialog.Builder(MainPrivacy.this)

    .setView(enterFileNameView).setNegativeButton("取消", null)
                    .setPositiveButton("确定", new
DialogInterface.OnClickListener() {

                        @Override
                        public void onClick(DialogInterface dialog,
int which) {
                            DBHelper helper = new
DBHelper(MainPrivacy.this);
                            fileName =
et_enter_file_name.getText().toString();
                            if(helper.findFile(fileName)){
                                new
AlertDialog.Builder(MainPrivacy.this).setTitle("请选择方式")
                                    .setPositiveButton("覆盖", new
DialogInterface.OnClickListener() {

                                        @Override
                                        public void
onClick(DialogInterface dialog, int which) {
                                            DBHelper helper = new
```

```
DBHelper(MainPrivacy.this);
                                        helper.deleteAll(1);

    helper.restoreData(fileName);
                                        list =
helper.getAllUser(privacy);
                                        adapter = new
SimpleAdapter(MainPrivacy.this,
                                                list,
                                                R.layout.listitem,
                                                new
String[]{"imageid","name","mobilephone"},
                                                new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});
                                        LayoutInflater li =
LayoutInflater.from(MainPrivacy.this);
                                        View backup_view =
li.inflate(R.layout.backup_progress, null);
                                        progressDialog =  new
AlertDialog.Builder(MainPrivacy.this)
                                        .setTitle("正在还原数据...")
                                        .setView(backup_view)
                                        .create();
                                        progressDialog.show();
                                        ProgressBar bar =
(ProgressBar) backup_view.findViewById(R.id.pb_backup);

                                        Button btn_backup_ok =
(Button)backup_view.findViewById(R.id.btn_backuup_ok);
                                        bar.setMax(list.size());
                                        for(int
i=0;i<=list.size();i++) {
                                            bar.setProgress(i);
                                        }
                                        progressDialog.setTitle("还
原完成！一共还原了 "+ list.size() + " 条记录");

    btn_backup_ok.setVisibility(View.VISIBLE);

    btn_backup_ok.setOnClickListener(new OnClickListener() {

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

```
    progressDialog.dismiss();

    mainMenuDialog.dismiss();

    lv.setAdapter(adapter);

                                            }

                                        });
                                    }
                                })
                                .setNegativeButton("添加", new
DialogInterface.OnClickListener() {

                                    @Override
                                    public void
onClick(DialogInterface dialog, int which) {
                                        DBHelper helper = new
DBHelper(MainPrivacy.this);
                                        int preNum = list.size();

    helper.restoreData(fileName);
                                        list =
helper.getAllUser(privacy);
                                        adapter = new
SimpleAdapter(MainPrivacy.this,
                                                list,
                                                R.layout.listitem,
                                                new
String[]{"imageid","name","mobilephone"},
                                                new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});
                                        LayoutInflater li =
LayoutInflater.from(MainPrivacy.this);
                                        View backup_view =
li.inflate(R.layout.backup_progress, null);
                                        progressDialog = new
AlertDialog.Builder(MainPrivacy.this)
                                        .setTitle("正在还原数据...")
                                        .setView(backup_view)
                                        .create();
                                        progressDialog.show();
                                        ProgressBar bar =
```

```
(ProgressBar) backup_view.findViewById(R.id.pb_backup);
                                                Button btn_backup_ok =
(Button)backup_view.findViewById(R.id.btn_backuup_ok);
                                                bar.setMax(list.size());
                                                for(int
i=0;i<=list.size();i++) {
                                                    bar.setProgress(i);
                                                }
                                                progressDialog.setTitle("还
原完成！一共还原了 "+ (list.size()-preNum) + " 条记录");
    btn_backup_ok.setVisibility(View.VISIBLE);
    btn_backup_ok.setOnClickListener(new OnClickListener() {
                                                    @Override
                                                    public void onClick(View
v) {
    progressDialog.dismiss();
    mainMenuDialog.dismiss();
                                                        if(list.size() != 0)
{
    mainLinearLayout.setBackgroundDrawable(null);
                                                        }
    lv.setAdapter(adapter);
                                                    }
                                                });
                                            }
                                        })
                                        .setNeutralButton("取消", new
DialogInterface.OnClickListener() {
                                            @Override
                                            public void
onClick(DialogInterface dialog, int which) {
                                            }
                                        }).create().show();
```

```
                                } else {

    Toast.makeText(enterFileNameDialog.getContext(), "找不到备份文件",
Toast.LENGTH_LONG).show();
                                }
                            }
                        })
                        .create();
                        et_enter_file_name =
(EditText)enterFileNameView.findViewById(R.id.et_enter_file_name);
                        et_enter_file_name.setText("priv_data");
                        et_enter_file_name.requestFocus();
                        et_enter_file_name.selectAll();
                        enterFileNameDialog.show();


                         adapter = new SimpleAdapter(
                              MainPrivacy.this,
                              list,
                              R.layout.listitem,
                              new
String[]{"imageid","name","mobilephone"},
                              new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});

                         lv.setAdapter(adapter);//显示所有数据
                        mainMenuDialog.dismiss();
                        break;
                    }
                    case 4:{
                        finish();
                        break;
                    }
                    case 5:{
                        mainMenuDialog.dismiss();
                        break;
                    }
                }

            }});
        }

    }
```

```
    private void loadSearchLinearout() {

        if(searchLinearout == null) {
            searchLinearout = (LinearLayout)
findViewById(R.id.ll_search);
            et_search = (EditText)findViewById(R.id.et_search);
            et_search.setOnKeyListener(new OnKeyListener(){
                @Override
                public boolean onKey(View arg0, int arg1, KeyEvent arg2)
{
                    String condition = et_search.getText().toString();
                    if(condition.equals("")) {
                        lv.setAdapter(adapter);
                    }
                    DBHelper helper = new DBHelper(MainPrivacy.this);
                    list = helper.getUsers(condition,privacy);
                    SimpleAdapter searchAdapter =
                        new SimpleAdapter(
                                    MainPrivacy.this,
                                    list,
                                    R.layout.listitem,
                                    new
String[]{"imageid","name","mobilephone"},
                                    new
int[]{R.id.user_image,R.id.tv_name,R.id.tv_mobilephone});
                    lv.setAdapter(searchAdapter);  //将整合好的adapter交给
listview，显示给用户看
                    if(list.size() == 0) {
                        Drawable nodata_bg =
getResources().getDrawable(R.drawable.nodata_bg);

    mainLinearLayout.setBackgroundDrawable(nodata_bg);
                        setTitle("没有查到任何数据");
                    } else {
                        setTitle( "共查到 " + list.size()+" 条记录");

                        mainLinearLayout.setBackgroundDrawable(null);
                    }
                    return false;
                }});

        }
```

```
    }

    private SimpleAdapter getMenuAdapter(String[] menuNameArray,
            int[] imageResourceArray) {
        ArrayList<HashMap<String, Object>> data = new
ArrayList<HashMap<String, Object>>();
        for (int i = 0; i < menuNameArray.length; i++) {
            HashMap<String, Object> map = new HashMap<String, Object>();
            map.put("itemImage", imageResourceArray[i]);
            map.put("itemText", menuNameArray[i]);
            data.add(map);
        }
        SimpleAdapter simperAdapter =
                new SimpleAdapter(
                                this,
                                data,
                                R.layout.item_menu,
                                new String[] { "itemImage", "itemText" },
                                new int[] { R.id.item_image,
R.id.item_text });
        return simperAdapter;
    }

    /**
     * 当退出的时候，回收资源
     */
    @Override
    protected void onDestroy() {
        if(confirmDialog != null) {
            confirmDialog = null;
        }
        if(mainMenuDialog != null) {
            mainMenuDialog = null;
        }
        if(searchLinearout != null) {
            searchLinearout = null;
        }
        if(mainMenuView != null) {
            mainMenuView = null;
        }
        if(mainMenuGrid != null) {
            mainMenuGrid = null;
        }
        if(bottomMenuGrid != null) {
```

```
            bottomMenuGrid = null;
        }
        if(adapter != null) {
            adapter = null;
        }
        if(list != null) {
            list = null;
        }
        if(lv != null) {
            lv = null;
        }

        Toast.makeText(this, "退出秘密仓库", Toast.LENGTH_LONG).show();
        System.out.println("destory!!!");
        super.onDestroy();
    }

}
```

3.4 UserDetail.java

```
public class UserDetail extends Activity implements ViewFactory {

    EditText et_name;
    EditText et_mobilePhone;
    EditText et_officePhone;
    EditText et_familyPhone;
    EditText et_position;
    EditText et_company;
    EditText et_address;
    EditText et_zipCode;
    EditText et_otherContact;
    EditText et_email;
    EditText et_remark;

    Button btn_save;
    Button btn_return;
    Button btn_delete;
    //头像的按钮
    ImageButton imageButton;
    //用flag来判断按钮的状态   false表示查看点击修改状态  true表示点击修改保存
状态
    boolean flag = false;
    boolean imageChanged = false;
```

```
boolean isDataChanged = false;

int currentImagePosition;
int previousImagePosition;

String[] callData;
//表示状态：打电话，发短信，发邮件
String status;
//拥有一个user实例，这个对象由Intent传过来
User user;
Gallery gallery;
ImageSwitcher is;

View numChooseView;
View imageChooseView;

//号码选择的对话框
AlertDialog numChooseDialog;
AlertDialog imageChooseDialog;
/**
 * 所有的图像图片
 */
private  int[] images
        = new int[]{R.drawable.icon
    ,R.drawable.image1,R.drawable.image2,R.drawable.image3
    ,R.drawable.image4,R.drawable.image5,R.drawable.image6
    ,R.drawable.image7,R.drawable.image8,R.drawable.image9
    ,R.drawable.image10,R.drawable.image11,R.drawable.image12
    ,R.drawable.image13,R.drawable.image14,R.drawable.image15
    ,R.drawable.image16,R.drawable.image17,R.drawable.image18
    ,R.drawable.image19,R.drawable.image20,R.drawable.image21
    ,R.drawable.image22,R.drawable.image23,R.drawable.image24
    ,R.drawable.image25,R.drawable.image26,R.drawable.image27
    ,R.drawable.image28,R.drawable.image29,R.drawable.image30};



@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.userdetail);

    //获得意图
    Intent intent = getIntent();
```

```
//从意图中得到需要的user对象
user = (User) intent.getSerializableExtra("user");
// 加载数据,往控件上赋值
loadUserData();
// 设置EditText不可编辑
setEditTextDisable();

//为按钮添加监听类
btn_save.setOnClickListener(new OnClickListener(){
    @Override
    public void onClick(View arg0) {
        if(!flag) {
            btn_save.setText("保存修改");
            setEditTextAble();
            flag = true;
        } else {
            //往数据库里面更新数据
            setTitle("modify");
            modify();
            setEditTextDisable();
            setColorToWhite();
            btn_save.setText("修改");
            flag = false;
        }

    }});


btn_return.setOnClickListener(new OnClickListener(){

    @Override
    public void onClick(View v) {
        if(isDataChanged) {
            setResult(4);
        }  else {
            setResult(5);
        }
        finish();
    }});


btn_delete.setOnClickListener(new OnClickListener(){

    @Override
```

```
            public void onClick(View v) {
                new AlertDialog.Builder(UserDetail.this).
                setPositiveButton("确定",new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which)
{
                        delete();
                        setResult(4);
                        finish();
                    }
                }).setNegativeButton("取消", new
DialogInterface.OnClickListener() {

                    @Override
                    public void onClick(DialogInterface dialog, int which)
{

                    }
                }).setTitle("是否要删除?").create().show();

            }});


        imageButton.setOnClickListener(new OnClickListener(){

            @Override
            public void onClick(View v) {
                loadImage();//加载imageChooseView，只加载一次
                initImageChooseDialog();//加载imageChooseDialog，只加载一
次
                imageChooseDialog.show();


            }});




    }

    /**
     * 获得布局文件中的控件，并且根据传递过来user对象对控件进行赋值
     */
    public void loadUserData() {
```

```
    // 获得EditText控件
    et_name = (EditText) findViewById(R.id.username);
    et_mobilePhone = (EditText) findViewById(R.id.mobilephone);
    et_officePhone = (EditText) findViewById(R.id.officephone);
    et_familyPhone = (EditText) findViewById(R.id.familyphone);
    et_position = (EditText) findViewById(R.id.position);
    et_company = (EditText) findViewById(R.id.company);
    et_address = (EditText) findViewById(R.id.address);
    et_zipCode = (EditText) findViewById(R.id.zipcode);
    et_otherContact = (EditText) findViewById(R.id.othercontact);
    et_email = (EditText) findViewById(R.id.email);
    et_remark = (EditText) findViewById(R.id.remark);

    // 获得Button控件
    btn_save = (Button)findViewById(R.id.save);
    btn_return = (Button)findViewById(R.id.btn_return);
    btn_delete = (Button)findViewById(R.id.delete);
    imageButton = (ImageButton)findViewById(R.id.image_button);

    // 为控件赋值
    et_name.setText(user.username);
    et_mobilePhone.setText(user.mobilePhone);
    et_familyPhone.setText(user.familyPhone);
    et_officePhone.setText(user.officePhone);
    et_company.setText(user.company);
    et_address.setText(user.address);
    et_zipCode.setText(user.zipCode);
    et_otherContact.setText(user.otherContact);
    et_email.setText(user.email);
    et_remark.setText(user.remark);
    et_position.setText(user.position);
    imageButton.setImageResource(user.imageId);
}

/**
 * 设置EditText为不可用
 */
private void setEditTextDisable() {
    et_name.setEnabled(false);
    et_mobilePhone.setEnabled(false);
    et_officePhone.setEnabled(false);
    et_familyPhone.setEnabled(false);
    et_position.setEnabled(false);
    et_company.setEnabled(false);
```

```
    et_address.setEnabled(false);
    et_zipCode.setEnabled(false);
    et_otherContact.setEnabled(false);
    et_email.setEnabled(false);
    et_remark.setEnabled(false);
    imageButton.setEnabled(false);
    setColorToWhite();

}

/**
 * 设置EditText为可用状态
 */
private void setEditTextAble() {
    et_name.setEnabled(true);
    et_mobilePhone.setEnabled(true);
    et_officePhone.setEnabled(true);
    et_familyPhone.setEnabled(true);
    et_position.setEnabled(true);
    et_company.setEnabled(true);
    et_address.setEnabled(true);
    et_zipCode.setEnabled(true);
    et_otherContact.setEnabled(true);
    et_email.setEnabled(true);
    et_remark.setEnabled(true);
    imageButton.setEnabled(true);
    setColorToBlack();
}

/**
 *  设置显示的字体颜色为黑色
 */
private void setColorToBlack() {

    et_name.setTextColor(Color.BLACK);
    et_mobilePhone.setTextColor(Color.BLACK);
    et_officePhone.setTextColor(Color.BLACK);
    et_familyPhone.setTextColor(Color.BLACK);
    et_position.setTextColor(Color.BLACK);
    et_company.setTextColor(Color.BLACK);
    et_address.setTextColor(Color.BLACK);
    et_zipCode.setTextColor(Color.BLACK);
    et_otherContact.setTextColor(Color.BLACK);
    et_email.setTextColor(Color.BLACK);
```

```
    et_remark.setTextColor(Color.BLACK);
}

/**
 *  设置显示的字体颜色为白色
 */
private void setColorToWhite() {
    et_name.setTextColor(Color.WHITE);
    et_mobilePhone.setTextColor(Color.WHITE);
    et_officePhone.setTextColor(Color.WHITE);
    et_familyPhone.setTextColor(Color.WHITE);
    et_position.setTextColor(Color.WHITE);
    et_company.setTextColor(Color.WHITE);
    et_address.setTextColor(Color.WHITE);
    et_zipCode.setTextColor(Color.WHITE);
    et_otherContact.setTextColor(Color.WHITE);
    et_email.setTextColor(Color.WHITE);
    et_remark.setTextColor(Color.WHITE);
}

/**
 * 获得最新数据，创建DBHelper对象，更新数据库
 */
private void modify() {
    user.username = et_name.getText().toString();
    user.address = et_address.getText().toString();
    user.company = et_company.getText().toString();
    user.email = et_email.getText().toString();
    user.familyPhone = et_familyPhone.getText().toString();
    user.mobilePhone = et_mobilePhone.getText().toString();
    user.officePhone = et_officePhone.getText().toString();
    user.otherContact = et_otherContact.getText().toString();
    user.position = et_position.getText().toString();
    user.remark = et_remark.getText().toString();
    user.zipCode = et_zipCode.getText().toString();
    if(imageChanged) {
        user.imageId = images[currentImagePosition%images.length];
    }

    DBHelper helper = new DBHelper(this);
    //打开数据库
    helper.openDatabase();
    helper.modify(user);
    isDataChanged = true;
```

```
}

private void delete() {
    DBHelper helper = new DBHelper(this);
    //打开数据库
    helper.openDatabase();
    helper.delete(user._id);
}

/**
 * 为Menu添加几个选项
 */
@Override
public boolean onCreateOptionsMenu(Menu menu) {

    menu.addSubMenu(0, Menu.FIRST, 1, "打电话");
    menu.addSubMenu(0, Menu.FIRST+1, 2, "发短信");
    menu.addSubMenu(0, Menu.FIRST+2, 3, "发邮件");

    //为每一个Item设置图标
    MenuItem item = menu.getItem(Menu.FIRST-1);
    item.setIcon(R.drawable.dial);
    MenuItem item1 = menu.getItem(Menu.FIRST);
    item1.setIcon(R.drawable.send_sms);
    MenuItem item2 = menu.getItem(Menu.FIRST+1);
    item2.setIcon(R.drawable.mail);
    return super.onCreateOptionsMenu(menu);
}

/**
 * 为每一个MenuItem添加事件
 */
@Override
public boolean onMenuItemSelected(int featureId, MenuItem item) {

    switch(item.getItemId()){
        case Menu.FIRST: {
            //将状态设置为打电话
            status = Intent.ACTION_CALL;
            if(callData == null) {
                //加载可用的号码
                loadAvailableCallData();
            }
```

```
                if(callData.length == 0) {
                    //提示没有可用的号码
                    Toast.makeText(this, "没有可用的号码！",
Toast.LENGTH_LONG).show();
                } else if(callData.length == 1) {
                    //如果之有一个可用的号码，这直接使用这个号码拨出
                    Intent intent =
                        new Intent(Intent.ACTION_CALL,Uri.parse("tel://"
+ callData[0]));
                    startActivity(intent);
                } else {
                    //如果有2个或者2个以上号码，弹出号码选择对话框
                    initNumChooseDialog();
                }
                break;
            }
            case Menu.FIRST+1: {
                status = Intent.ACTION_SENDTO;
                if(callData == null) {
                    loadAvailableCallData();
                }
                if(callData.length == 0) {
                    //提示没有可用的号码
                    Toast.makeText(this, "没有可用的号码！",
Toast.LENGTH_LONG).show();
                } else if(callData.length == 1) {
                    //如果之后又一个可用的号码，这直接使用这个号码拨出
                    Intent intent =
                        new
Intent(Intent.ACTION_SENDTO,Uri.parse("smsto://" + callData[0]));
                    startActivity(intent);
                } else {
                    initNumChooseDialog();
                }
                break;
            }
            case Menu.FIRST+2: {

                if(user.email.equals("")) {
                    Toast.makeText(this, "没有可用的邮箱！",
Toast.LENGTH_LONG).show();
                } else {
                    Uri emailUri = Uri.parse("mailto:" + user.email);
                    Intent intent = new Intent(Intent.ACTION_SENDTO,
```

```
emailUri);
                    startActivity(intent);
                }
                break;
            }

        }

        return super.onMenuItemSelected(featureId, item);
    }
    /**
     * 装载头像
     */
    public void loadImage() {
        if(imageChooseView == null) {
            LayoutInflater li = LayoutInflater.from(UserDetail.this);
            imageChooseView = li.inflate(R.layout.imageswitch, null);
            gallery =
(Gallery)imageChooseView.findViewById(R.id.gallery);
            gallery.setAdapter(new ImageAdapter(this));
            gallery.setSelection(images.length/2);
            is =
(ImageSwitcher)imageChooseView.findViewById(R.id.imageswitch);
            is.setFactory(this);
            gallery.setOnItemSelectedListener(new
OnItemSelectedListener(){

                @Override
                public void onItemSelected(AdapterView<?> arg0, View arg1,
                        int arg2, long arg3) {
                    // TODO Auto-generated method stub
                    currentImagePosition = arg2 % images.length;
                    is.setImageResource(images[arg2 % images.length]);

                }

                @Override
                public void onNothingSelected(AdapterView<?> arg0) {

                }});
        }

    }
```

```
    public void initNumChooseDialog() {
        if(numChooseDialog == null) {
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            LayoutInflater inflater = LayoutInflater.from(this);
            numChooseView = inflater.inflate(R.layout.numchoose, null);
            ListView lv =
(ListView)numChooseView.findViewById(R.id.num_list);
            ArrayAdapter array =
                new
ArrayAdapter(this,android.R.layout.simple_list_item_1,callData);
            lv.setAdapter(array);
            lv.setOnItemClickListener(new OnItemClickListener(){
                @Override
                public void onItemClick(AdapterView<?> arg0, View arg1,
int arg2,
                        long arg3) {
                    String num =
String.valueOf(arg0.getItemAtPosition(arg2));
                    Intent intent = null;
                    if(status.equals(Intent.ACTION_CALL)) {
                        intent = new
Intent(Intent.ACTION_CALL,Uri.parse("tel://" + num));
                    } else {
                        intent = new
Intent(Intent.ACTION_SENDTO,Uri.parse("smsto://" + num));
                    }

                    startActivity(intent);
                    //对话框消失
                    numChooseDialog.dismiss();
                }});


            builder.setView(numChooseView);
            numChooseDialog = builder.create();

        }
        numChooseDialog.show();
    }

    public void initImageChooseDialog() {
        if(imageChooseDialog == null) {
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setTitle("请选择图像")
```

```
            .setView(imageChooseView).setPositiveButton("确定", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    imageChanged = true;
                    previousImagePosition = currentImagePosition;

    imageButton.setImageResource(images[currentImagePosition%images.l
ength]);
                }
            })
            .setNegativeButton("取消", new
DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface dialog, int which) {
                    currentImagePosition = previousImagePosition;


                }
            });
            imageChooseDialog = builder.create();
        }
    }
    /**
     * 装载可用的号码
     */
    public void loadAvailableCallData() {
        ArrayList<String> callNums = new ArrayList<String>();
        if(!user.mobilePhone.equals("")) {
            callNums.add(user.mobilePhone);
        }
        if(!user.familyPhone.equals("")) {
            callNums.add(user.familyPhone);
        }

        if(!user.officePhone.equals("")) {
            callNums.add(user.officePhone);
        }


        callData = new String[callNums.size()];

        for(int i=0;i<callNums.size();i++) {
            callData[i] = callNums.get(i);
```

```
        }


    }


    /**
     * 自定义头像适配器
     * @author Administrator
     *
     */
    class ImageAdapter extends BaseAdapter {

        private Context context;

        public ImageAdapter(Context context) {
            this.context = context;
        }

        @Override
        public int getCount() {
            // TODO Auto-generated method stub
            return Integer.MAX_VALUE;
        }

        @Override
        public Object getItem(int position) {
            // TODO Auto-generated method stub
            return position;
        }

        @Override
        public long getItemId(int position) {
            // TODO Auto-generated method stub
            return position;
        }


        /**
         * gallery从这个方法中拿到image
         */
        @Override
        public View getView(int position, View convertView, ViewGroup
parent) {
```

```
        ImageView iv = new ImageView(context);
        iv.setImageResource(images[position%images.length]);
        iv.setAdjustViewBounds(true);
        iv.setLayoutParams(new Gallery.LayoutParams(80,80));
        iv.setPadding(15, 10, 15, 10);
        return iv;
    }

}

@Override
public View makeView() {
    ImageView view = new ImageView(this);
    view.setBackgroundColor(0xff000000);
    view.setScaleType(ScaleType.FIT_CENTER);
    view.setLayoutParams(new ImageSwitcher.LayoutParams(90,90));
    return view;
}
/**
 * 当退出的时候，回收资源
 */
@Override
protected void onDestroy() {
    if(is != null) {
        is = null;
    }
    if(gallery != null) {
        gallery = null;
    }
    if(imageChooseDialog != null) {
        imageChooseDialog = null;
    }
    if(imageChooseView != null) {
        imageChooseView = null;
    }
    if(imageButton != null) {
        imageButton = null;
    }
    if(numChooseDialog != null) {
        numChooseDialog = null;
    }
    if(numChooseView != null) {
        numChooseView = null;
    }
```

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

3.5 DBHelper.java

```
public class DBHelper {

    public static final String DB_DBNAME="contact";

    public static final String DB_TABLENAME="user";

    public static final int VERSION = 4;

    public static SQLiteDatabase dbInstance;

    private MyDBHelper myDBHelper;

    private StringBuffer tableCreate;

    private Context context;

    public DBHelper(Context context) {
        this.context = context;
    }

    public void openDatabase() {
        if(dbInstance == null) {
            myDBHelper = new MyDBHelper(context,DB_DBNAME,VERSION);
            dbInstance = myDBHelper.getWritableDatabase();
        }
    }
    /**
     * 往数据库里面的user表插入一条数据，若失败返回-1
     * @param user
     * @return   失败返回-1
     */
    public long insert(User user) {
        ContentValues values = new ContentValues();
        values.put("name", user.username);
        values.put("mobilephone", user.mobilePhone);
        values.put("officephone", user.officePhone);
        values.put("familyphone", user.familyPhone);
```

```
        values.put("address", user.address);
        values.put("othercontact", user.otherContact);
        values.put("email",user.email);
        values.put("position", user.position);
        values.put("company", user.company);
        values.put("zipcode", user.zipCode);
        values.put("remark", user.remark);
        values.put("imageid", user.imageId);
        values.put("privacy", user.privacy);
        return dbInstance.insert(DB_TABLENAME, null, values);
    }

    /**
     * 获得数据库中所有的用户，将每一个用户放到一个map中去，然后再将map放到list里面去返回
     * @param privacy
     * @return list
     */

    public ArrayList getAllUser(boolean privacy) {
        ArrayList list = new ArrayList();
        Cursor cursor = null;
        if(privacy) {
            cursor = dbInstance.query(DB_TABLENAME,
                    new String[]{"_id","name","mobilephone","officephone","familyphone","address","othercontact","email","position","company","zipcode","remark","imageid"},
                    "privacy=1",
                    null,
                    null,
                    null,
                    null);
        } else {
            cursor = dbInstance.query(DB_TABLENAME,
                    new String[]{"_id","name","mobilephone","officephone","familyphone","address","othercontact","email","position","company","zipcode","remark","imageid"},
                    "privacy=0",
                    null,
                    null,
                    null,
                    null);
```

```
        }


        while(cursor.moveToNext()) {
            HashMap item = new HashMap();
            item.put("_id",
cursor.getInt(cursor.getColumnIndex("_id")));
            item.put("name",
cursor.getString(cursor.getColumnIndex("name")));
            item.put("mobilephone",
cursor.getString(cursor.getColumnIndex("mobilephone")));
            item.put("officephone",
cursor.getString(cursor.getColumnIndex("officephone")));
            item.put("familyphone",
cursor.getString(cursor.getColumnIndex("familyphone")));
            item.put("address",
cursor.getString(cursor.getColumnIndex("address")));
            item.put("othercontact",
cursor.getString(cursor.getColumnIndex("othercontact")));
            item.put("email",
cursor.getString(cursor.getColumnIndex("email")));
            item.put("position",
cursor.getString(cursor.getColumnIndex("position")));
            item.put("company",
cursor.getString(cursor.getColumnIndex("company")));
            item.put("zipcode",
cursor.getString(cursor.getColumnIndex("zipcode")));
            item.put("remark",
cursor.getString(cursor.getColumnIndex("remark")));
            item.put("imageid",
cursor.getInt(cursor.getColumnIndex("imageid")));
            list.add(item);
        }

        return list;
    }


    public void modify(User user) {
        ContentValues values = new ContentValues();
        values.put("name", user.username);
        values.put("mobilephone", user.mobilePhone);
        values.put("officephone", user.officePhone);
        values.put("familyphone", user.familyPhone);
```

```
        values.put("address", user.address);
        values.put("othercontact", user.otherContact);
        values.put("email",user.email);
        values.put("position", user.position);
        values.put("company", user.company);
        values.put("zipcode", user.zipCode);
        values.put("remark", user.remark);
        values.put("imageid", user.imageId);

        dbInstance.update(DB_TABLENAME, values, "_id=?", new
String[]{String.valueOf(user._id)});
    }

    public void delete(int _id) {
        dbInstance.delete(DB_TABLENAME, "_id=?", new
String[]{String.valueOf(_id)});
    }
    public void deleteAll(int privacy) {
        dbInstance.delete(DB_TABLENAME, "privacy=?", new
String[]{String.valueOf(privacy)});
    }

    public int getTotalCount() {
        Cursor cursor = dbInstance.query(DB_TABLENAME, new
String[]{"count(*)"}, null, null, null, null, null);
        cursor.moveToNext();
        return cursor.getInt(0);
    }


    public ArrayList getUsers(String condition, boolean privacy) {
        ArrayList list = new ArrayList();
        String strSelection = "";
        if(privacy) {
            strSelection = "and privacy = 1";
        } else {
            strSelection = "and privacy = 0";
        }
        String sql = "select * from " + DB_TABLENAME + " where 1=1 and (name
like '%" + condition + "%' " +
                "or mobilephone like '%" + condition + "%' or familyphone
like '%" + condition + "%' " +
                    "or officephone like '%" + condition + "%')" +
strSelection;
```

```
        Cursor cursor = dbInstance.rawQuery(sql, null);
        while(cursor.moveToNext()) {
            HashMap item = new HashMap();
item.put("_id", cursor.getInt(cursor.getColumnIndex("_id")));
item.put("name", cursor.getString(cursor.getColumnIndex("name")));
item.put("mobilephone",
cursor.getString(cursor.getColumnIndex("mobilephone")));
item.put("officephone",
cursor.getString(cursor.getColumnIndex("officephone")));
item.put("familyphone",
cursor.getString(cursor.getColumnIndex("familyphone")));
item.put("address",
cursor.getString(cursor.getColumnIndex("address")));
item.put("othercontact",
cursor.getString(cursor.getColumnIndex("othercontact")));
item.put("email", cursor.getString(cursor.getColumnIndex("email")));
item.put("position",
cursor.getString(cursor.getColumnIndex("position")));
item.put("company",
cursor.getString(cursor.getColumnIndex("company")));
item.put("zipcode",
cursor.getString(cursor.getColumnIndex("zipcode")));
item.put("remark",
cursor.getString(cursor.getColumnIndex("remark")));
item.put("imageid", cursor.getInt(cursor.getColumnIndex("imageid")));
list.add(item);
        }
        return list;
    }

    public void deleteMarked(ArrayList<Integer> deleteId) {
        StringBuffer  strDeleteId = new StringBuffer();
        strDeleteId.append("_id=");
        for(int i=0;i<deleteId.size();i++) {
            if(i!=deleteId.size()-1) {
                strDeleteId.append(deleteId.get(i) + " or _id=");
            } else {
                strDeleteId.append(deleteId.get(i));
            }


        }
        dbInstance.delete(DB_TABLENAME, strDeleteId.toString(), null);
        System.out.println(strDeleteId.toString());
```

```
    }

    public void backupData(boolean privacy) {
        StringBuffer sqlBackup = new StringBuffer();
        Cursor cursor = null;
        if(privacy) {
            cursor = dbInstance.query(DB_TABLENAME,
                    new
String[]{"_id","name","mobilephone","officephone","familyphone","addr
ess","othercontact","email","position","company","zipcode","remark","
imageid,privacy"},
                    "privacy=1",
                    null,
                    null,
                    null,
                    null);
        } else {
            cursor = dbInstance.query(DB_TABLENAME,
                    new
String[]{"_id","name","mobilephone","officephone","familyphone","addr
ess","othercontact","email","position","company","zipcode","remark","
imageid,privacy"},
                    "privacy=0",
                    null,
                    null,
                    null,
                    null);
        }


        while(cursor.moveToNext()) {
            sqlBackup.append("insert into " + DB_TABLENAME +
"(name,mobilephone,officephone,familyphone,address,othercontact,email,
position,company,zipcode,remark,imageid,privacy)")
            .append(" values ('")

    .append(cursor.getString(cursor.getColumnIndex("name"))).append("
','")

    .append(cursor.getString(cursor.getColumnIndex("mobilephone"))).a
ppend("','")

    .append(cursor.getString(cursor.getColumnIndex("officephone"))).a
```

```
ppend("','")

    .append(cursor.getString(cursor.getColumnIndex("familyphone"))).a
ppend("','")

    .append(cursor.getString(cursor.getColumnIndex("address"))).appen
d("','")

    .append(cursor.getString(cursor.getColumnIndex("othercontact"))).
append("','")

    .append(cursor.getString(cursor.getColumnIndex("email"))).append(
"','")

    .append(cursor.getString(cursor.getColumnIndex("position"))).appe
nd("','")

    .append(cursor.getString(cursor.getColumnIndex("company"))).appen
d("','")

    .append(cursor.getString(cursor.getColumnIndex("zipcode"))).appen
d("','")

    .append(cursor.getString(cursor.getColumnIndex("remark"))).append
("',")

    .append(cursor.getInt(cursor.getColumnIndex("imageid"))).append(",
")
            .append(cursor.getInt(cursor.getColumnIndex("privacy")))
            .append(");").append("\n");
        }
        saveDataToFile(sqlBackup.toString(),privacy);

    }


    private void saveDataToFile(String strData,boolean privacy) {
        String fileName = "";
        if(privacy) {
            fileName = "priv_data.bk";
        } else {
            fileName = "comm_data.bk";
        }
        try {
```

```
        String SDPATH = Environment.getExternalStorageDirectory() + "/";
        File fileParentPath = new File(SDPATH + "zpContactData/");
        fileParentPath.mkdirs();
        File file = new File(SDPATH + "zpContactData/" + fileName);
        System.out.println("the file previous path = " +
file.getAbsolutePath());

        file.createNewFile();
        System.out.println("the file next path = " +
file.getAbsolutePath());
        FileOutputStream fos = new FileOutputStream(file);

        fos.write(strData.getBytes());
        fos.flush();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    public void restoreData(String fileName) {
        try {
        String SDPATH = Environment.getExternalStorageDirectory() + "/";
        File file = null;
        if(fileName.endsWith(".bk")) {
            file = new File(SDPATH + "zpContactData/"+ fileName);
        } else {
            file = new File(SDPATH + "zpContactData/"+ fileName + ".bk");
        }
        BufferedReader br = new BufferedReader(new FileReader(file));
        String str = "";
        while((str=br.readLine())!=null) {
            System.out.println(str);
            dbInstance.execSQL(str);
        }

        } catch(Exception e) {
            e.printStackTrace();
        }
    }

    public boolean findFile(String fileName) {
        String SDPATH = Environment.getExternalStorageDirectory() + "/";
        File file = null;
        if(fileName.endsWith(".bk")) {
            file = new File(SDPATH + "zpContact/"+fileName);
```

```
        } else {
            file = new File(SDPATH + "zpContact/"+fileName + ".bk");
        }

        if(file.exists()) {
            return true;
        } else {
            return false;
        }


}


class MyDBHelper extends SQLiteOpenHelper {

    public MyDBHelper(Context context, String name,
            int version) {
        super(context, name, null, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        tableCreate = new StringBuffer();
        tableCreate.append("create table ")
                   .append(DB_TABLENAME)
                   .append(" (")
                   .append("_id integer primary key autoincrement,")
                   .append("name text,")
                   .append("mobilephone text,")
                   .append("officephone text,")
                   .append("familyphone text,")
                   .append("address text,")
                   .append("othercontact text,")
                   .append("email text,")
                   .append("position text,")
                   .append("company text,")
                   .append("zipcode text,")
                   .append("remark text,")
                   .append("imageid int,")
                   .append("privacy int ")
                   .append(")");
        System.out.println(tableCreate.toString());
        db.execSQL(tableCreate.toString());
```

```
        }

        @Override
        public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
            String sql = "drop table if exists " + DB_TABLENAME;
            db.execSQL(sql);
            myDBHelper.onCreate(db);
        }

    }
```

3.6 User.java

```
public class User implements Serializable {

    public int _id;

    public String username;

    public String mobilePhone;

    public String officePhone;

    public String familyPhone;

    public String position;

    public String company;

    public String address;

    public String zipCode;

    public String email;

    public String otherContact;

    public String remark;

    public int imageId;

    public int privacy;// 1代表隐私用户 0代表普通用户
```

```
}
```

4. 配置文件

4.1 AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.zhengping.contact"
      android:versionCode="1"
      android:versionName="1.0">
    <application
    android:icon="@drawable/androidcontact"
    android:label="Contact"
    >
        <activity android:name=".Main"
                  android:label="Contact">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                  android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity  android:name=".UserDetail"/>
        <activity  android:name=".AddNew"/>
        <activity  android:name=".MainPrivacy"/>
    </application>
<uses-permission
android:name="android.permission.CALL_PHONE">
</uses-permission>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE">
</uses-permission>

</manifest>
```