

Lab3 实验报告

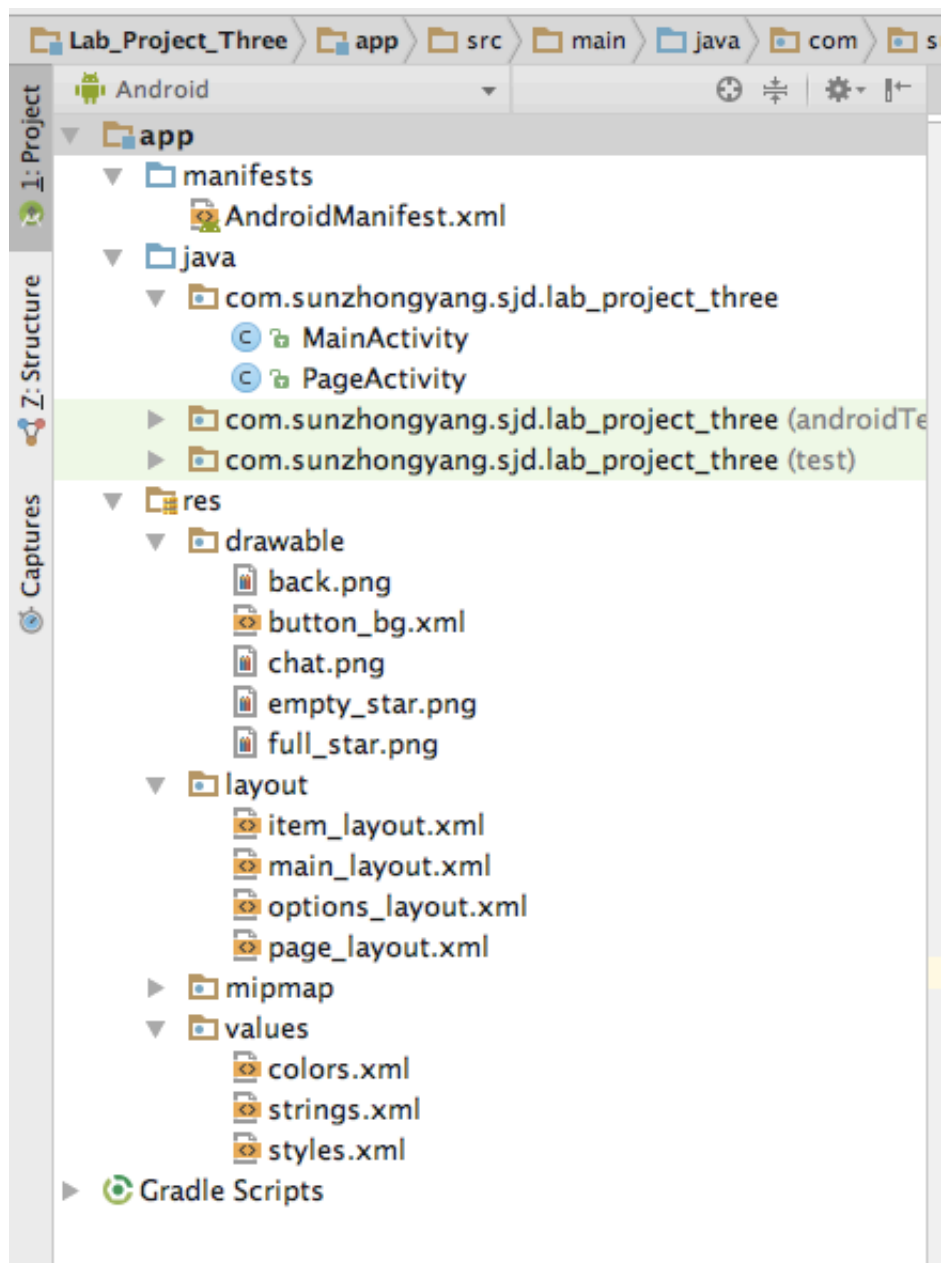
13331233 孙中阳

参考资料

《第一行代码 Android》 郭霖 著
《Lab3实验文档》

实验步骤

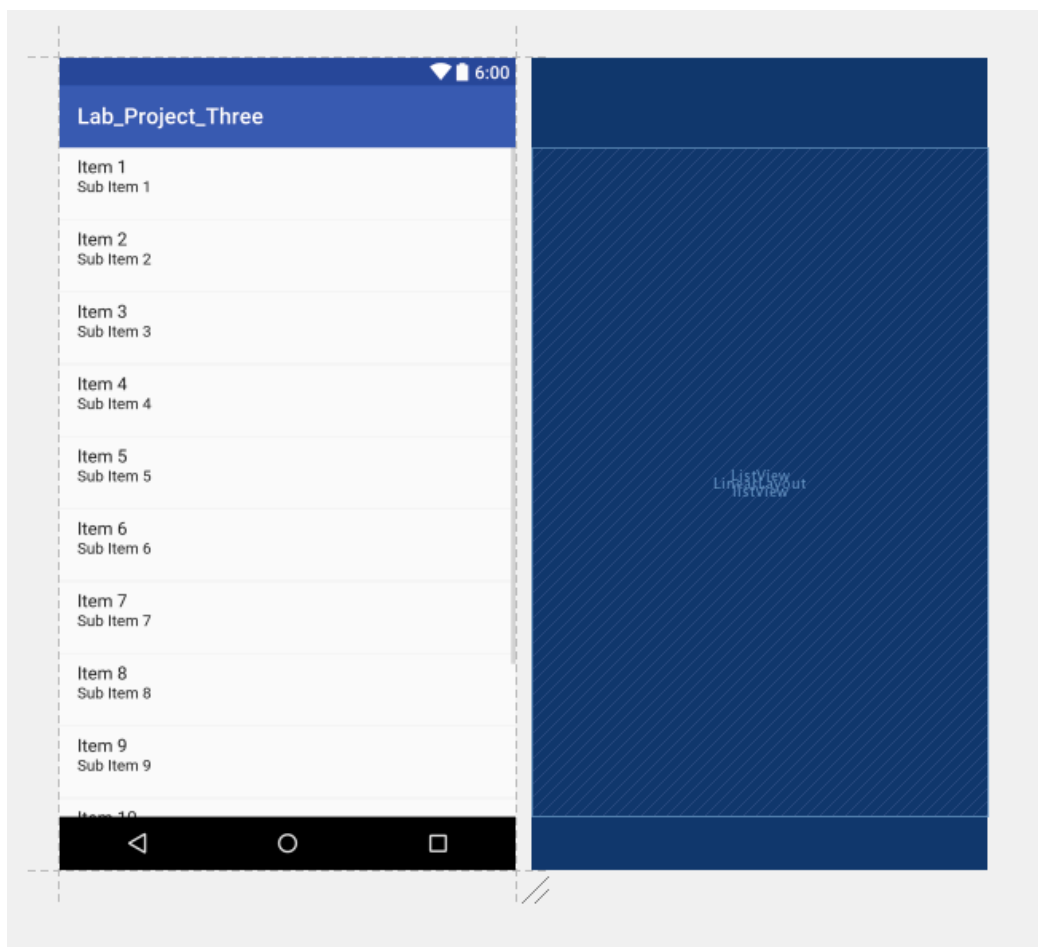
本次试验并不复杂，主要需要进一步熟悉Lab2中使用的基本控件，增进对布局的了解后在布局中应用 `ListView` 以显示表单数据，最终应用 `Intent` 及 `Bundle` 实现 `Activity` 的跳转和数据的传递 图为项目结构



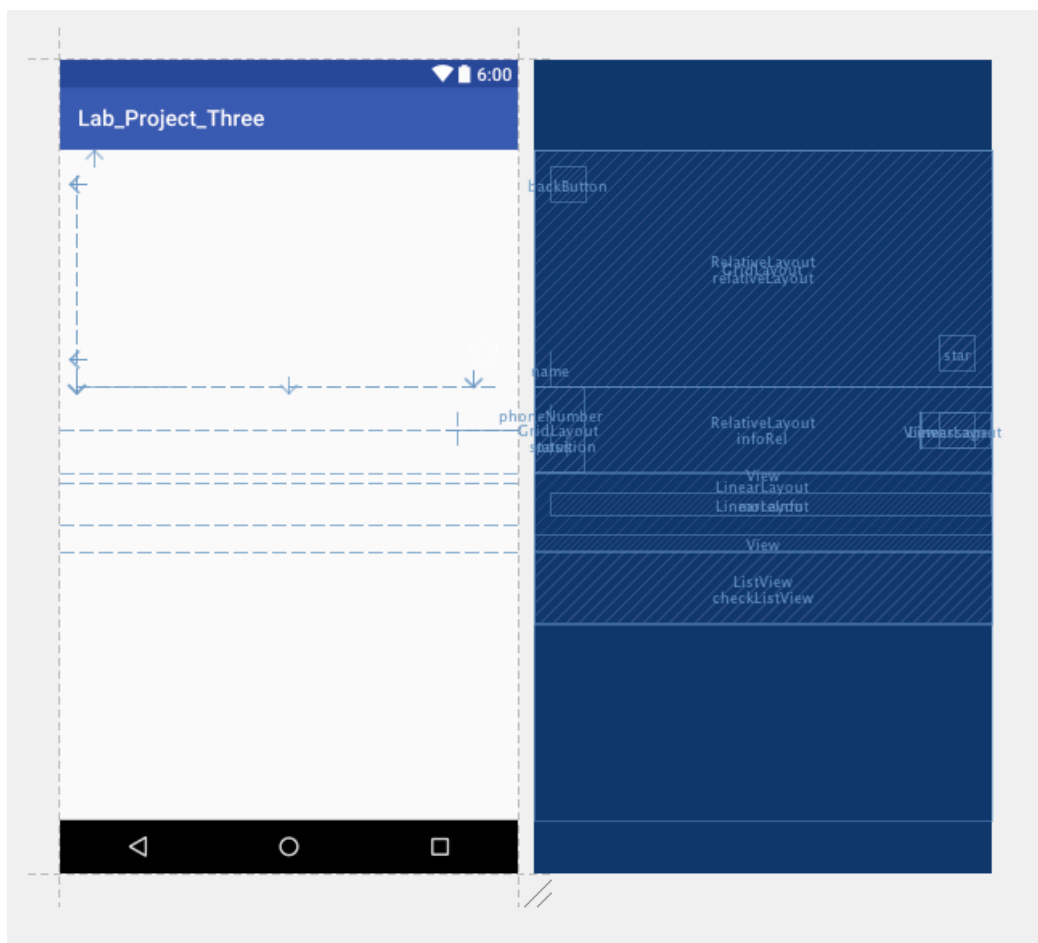
从实验文档看，需要实现的目标主要有

- 1.复习事件处理
- 2.学习 `Intent`、`Bundle` 在 `Activity` 跳转中的应用
- 3.学习 `ListView` 以及各类适配器的用法

图为主页面布局：



图为详情页布局：



接下来考虑实现

本次开发主要涉及三个方面：

1.主页面添加 `ListView` 为了在主页面添加 `ListView`，首先需要在其布局文件中加入一个 `ListView`

```
<ListView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/listView">
</ListView>
```

然后单独新建一个布局文件实现 `ListView` 的具体布局

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk
/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="17sp"
        android:layout_marginLeft="15dp"
        android:textColor="@color/black"
        android:layout_marginTop="17dp"
        android:layout_marginBottom="17dp"
        android:id="@+id/op" />

</LinearLayout>
```

这点和UWP有所不同，UWP会直接对 `ListView` 所在布局文件添加数据模板，以表示 `ListView` 内部的布局效果

之后在 `MainActivity.java` 中获取并初始化 `ListView`

```

//根据id获取ListView
ListView contactList = (ListView) findViewById(R.id.listView);

//构造一个List用于存储姓名首字母和姓名对
final List<Map<String, Object>> data = new ArrayList<>();
//构造一个字符串数组用于存储所有姓名
final String[] contactNames = new String[] {"Aaron", "Elvis",
"David", "Edwin", "Frank", "Joshua", "Ivan", "Mark", "Joseph", "Phoebe"};

//将姓名首字母和姓名的组合存入List
for(int i = 0; i < contactNames.length; i++)
{
    Map<String, Object> tmp = new LinkedHashMap<>();
    tmp.put("firstC", contactNames[i].charAt(0));
    tmp.put("contactN", contactNames[i]);
    data.add(tmp);
}

//构造一个SimpleAdapter,然后使用此Adapter初始化ListView
final SimpleAdapter sAdapter = new SimpleAdapter(this, data,
R.layout.item_layout, new String[] {"firstC", "contactN"},
new int[] {R.id.firstChar, R.id.contactName});
contactList.setAdapter(sAdapter);

```

这里使用了 `SimpleAdapter` 初始化 `ListView` , 没有实现自定义的构造器

最后为其注册两个事件监听器（短按，长按），中间需要注意的是删除联系人的方法

```

/////为ListView的每一个Item注册长按事件监听器
contactList.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener()
{
    @Override
    public boolean onItemLongClick(AdapterView<?> adapter,
View view, int i, long l)
    {
        final int count = i;

```

```

        //弹出一个提示框
        new AlertDialog.Builder(MainActivity.this)
            .setTitle("删除联系人")
            .setMessage("确定删除联系人" + data.get(count)
            ).get("contactN") + "?")
            //点击确定按钮则弹框提示确定按钮被按下
            .setPositiveButton("确定", new DialogInterface.OnClickListener()
            {
                @Override
                public void onClick(DialogInterface dialogInterface, int j)
                {
                    //从数据源删除有关被选中联系人的数据,并通知Adapter有关更改

                    data.remove(count);
                    sAdapter.notifyDataSetChanged();
                }
            })
            //点击取消按钮则提示取消按钮被按下
            .setNegativeButton("取消", new DialogInterface.OnClickListener()
            {
                @Override
                public void onClick(DialogInterface dialogInterface, int j)
                {
                }
            })
            .create()
            .show();

        return true;
    }
});

```

必须在 `ListView` 数据源的原地址删除数据，然后调用通知函数 `notifyDataSetChanged()`

2.构造一个 `Intent` 实现页面跳转并根据 `Bundle` 完成数据传递

Android 的导航和数据传递方法同样较UWP更为简便

只需要创建一个 `Intent` 并且使用创建的 `Intent` 来 `startActivity()` 即可

```
//构造一个Intent,携带被点击Item的姓名的信息创建一个联系人详情页
Intent intent = new Intent(MainActivity.this, PageActivity.class);
intent.putExtra("key", (String) data.get(i).get("contactN"));
startActivity(intent);
```

其中 `putExtra()` 部分负责放入额外数据,这里放入的是联系人的姓名,在新建的活动 `PageActivity` 中

```
//获取自上一页面传来的附加信息
Bundle bundle = getIntent().getExtras();
String name = bundle.getString("key");
```

通过 `Bundle` 获得附加的信息

3.一些页面的动态更改

本次试验要求页面根据传入的数据有所变化,同时有一定的自适应性 如最上部分占据页面三分之一的大小,背景色由传入联系人信息决定等

```
//根据id获取最上方带有背景色的相对布局
RelativeLayout rel = (RelativeLayout) findViewById(R.id.relativeLayout);

//根据联系人信息设置相对布局的颜色
rel.setBackgroundColor(Color.parseColor("#" + contacts.get(name)[3].toUpperCase()));

//将相对布局的高度设置为当前窗口高度的三分之一
rel.setLayoutParams().height = height / 3;
```

其他实现如收藏按钮 `Tag` 的使用已在文档中有所提示,背景色透明等实现细节也不多赘述,但在主页面设置首字母旁边的圆圈的实现上有所创新

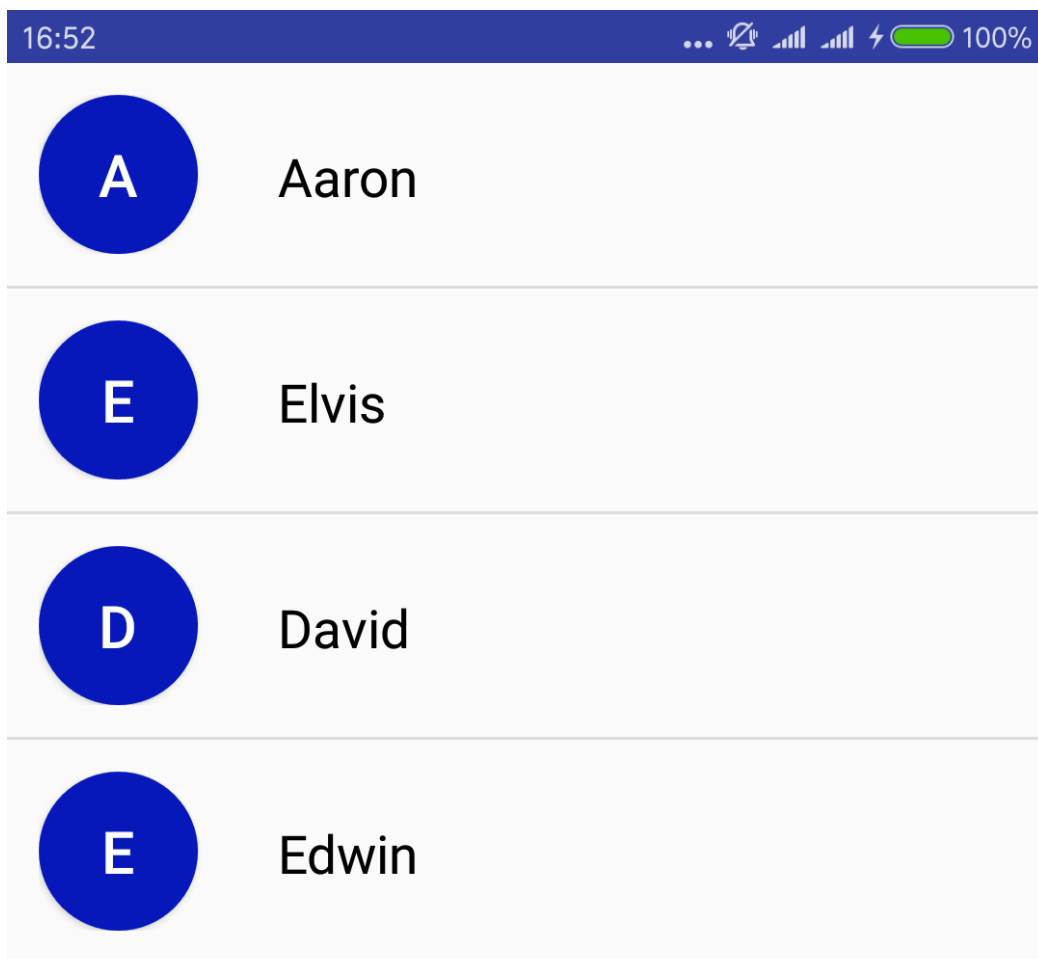

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/an
droid"
    android:shape="oval">




    <solid
        android:color="@color/first_character_color"/>
</shape>
```

这里并没有设置 `radius`，改为添加语句 `android:shape="oval"` 实现相同的效果，且不用担心 `radius` 的值过小导致最终出现的是圆角

实验成果截图

主页面





	Frank
	Joshua
	Ivan
	Mark


长按提示是否删除 Edwin


16:52

...












100%



Aaron



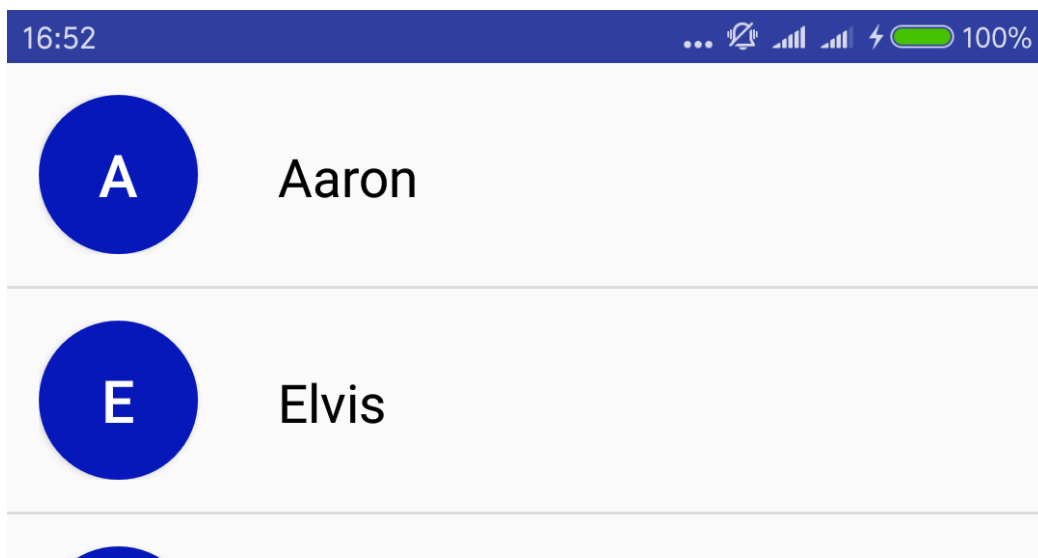
Elvis









David



删除 Edwin 之后的主页面



	David
	Frank
	Joshua
	Ivan
	Mark
	Joseph

短按打开 Frank 的详情页



Frank



13955188541

手机 安徽合肥移动



更多资料

编辑联系人

分享联系人

加入黑名单

删除联系人

点击收藏键图标改变

16:53



Frank



13955188541

手机 安徽合肥移动



更多资料

编辑联系人

分享联系人

加入黑名单

删除联系人

实验过程遇到的问题

这次同样遇到不少问题

1.首先是设置好 `Intent` 后跳转到详情页面会出现闪退

java中的内容都没有问题，布局文件也没有问题，最后查找资料发现漏掉了注册活动的部分

也就是需要在 `AndroidManifest` 中添加

```
<activity android:name=".PageActivity"
    >
</activity>
```

`Intent` 才能顺利传达

2.删除联系人出现闪退

一开始我删除联系人代码重写 `onClick()` 部分是这样的

```
public void onClick(DialogInterface dialogInterface, int i)
{
    //从数据源删除有关被选中联系人的数据,并通知Adapter有关更改
    data.remove(i);
    sAdapter.notifyDataSetChanged();
}
```

`onClick()` 参数中的 `i` 替换了 `setOnItemLongClickListener()` 参数中的 `i`，导致删除索引不正确，引发错误，后将其改名，问题得以解决

3.多次点击返回键可多次返回的问题

最初的实现中我使用创建一个新 `Intent` 的方式实现从详情页向主页面的跳转，后来发现创建新 `Intent` 后新的 `Activity` 会被压到活动栈的顶部，但前面的活动不会被销毁，于是可以从一个页面不断向前跳转，点击返回键可以从主页面跳到之前的详情页，这明显是不符合要求的

于是改为按返回键则销毁当前活动的方式，问题解决

```
//为返回键设置短按事件监听器
back.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    {
        //结束当前页面
        finish();
    }
});
```

布局过程中总是会出现一些小的问题，最终还算比较满意。在整个编辑过程中所有 `string` , `color` 值均保存在 `string.xml` , `color.xml` 中，这样也能避免一些错误

思考与总结

本次试验深入了解了Android开发的有关知识，对Android Studio的使用也有了更深刻的把握，尤其是和UI设计以及页面传递有关的内容。大部分内容可以通过查阅PPT或作业说明得到解决，只有少部分需要百度。Android较UWP应用范围更广，网上的资料也更多，问题也解决的比较顺利。总的来说很有收获。