Android应用软件设计

E 2 Intent and Intent Filter

学号: **SA18225112**

姓名: 何红飞

报告撰写时间:**2018/10/10**

1. 主题概述

1. 使用GridView替换已有的按钮导航

修改 E 2 的 SCOS 工程 MainScreen 布局，使用 GridView 替换已有的按钮导航： 点菜，查看订单，登录/注册，系统帮助;形成 launch board 导航样式

2. 结合使用tablelayout + viewPager + Fragment 完成页面

使用tablelayout 和viewPager完成要求的点餐页面，同时左右滑动切换Fragment

3. 使用Intent传递对象

使用Intent 来传递对象通常有两种实现方式，Serializable 和Parcelable。本应用采用了Serializable方式。

4. recyclerView 与 Adapter 实现滑动显示

使用 ListView 或 RecyclerView 加载当前类别的食物列表，并显示到当前屏幕。要求 ListView 或 RecyclerView 每个选项含有:菜名，价格，点菜按钮

5. Intent传递数值

可以在切换活动时，传递一些信息，以此作出不同的操作

2. 假设

在本应用中，我们需要对用户进行识别，区分出老客户和新客户，并且针对不同的客户结账时有不同的优惠政策。同时我们初步模拟完成了网上订餐的整个流程。

3. 实现和证明

1. Github：

https://github.com/WokeFei/SCOS.git

2. APK

3. 使用GridView重构主页导航：

首先对activity_main_screen.xml 重新布局，采用GridView，如图所示：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainScreen">

    <GridView
    android:id="@+id/gridView"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:numColumns="2">
    </GridView>

</LinearLayout>
```

然后在MainScreen中配置、加载布局。先创建两个数组：

```
// 定义两个数组
private int[] image = {R.drawable.ic_dish,R.drawable.ic_order,R.drawable.ic_user,R.drawable.ic_help};
private String[] text = {"点菜","订单","登录/注册","帮助"};
```

定义gridShow() 函数，使得grid导航栏显示出来，这里我们需要使用simpleAdapter封装数据，将图片显现出来。

```
// 定义gridShow（）函数，使得grid导航栏显示出来
private  void gridShow(){
    //显示GridView界面
    GridView gridView = (GridView) findViewById(R.id.gridView);
    ArrayList<HashMap<String,Object>> imagelist = new ArrayList<~>();
    //使用HashMap将图片添加到一个数组中
    for(int i = 0;i < image.length;i++){
        HashMap<String,Object> map = new HashMap<~>();
        map.put("image",image[i]);
        map.put("txt",text[i]);
        imagelist.add(map);
    }
    // 使用simpleAdapter封装数据，将图片显现出来
    SimpleAdapter simpleAdapter = new SimpleAdapter( context: this,imagelist,
            R.layout.items,new String[]{"image","txt"},new int[]{R.id.image_item,R.id.text_title});
    //设置GridView的适配器为新建的simpleAdapter
    gridView.setAdapter(simpleAdapter);
    //设置监听器
    gridView.setOnItemClickListener(this);
}
```

由于我们重构了导航栏，所以E2作业中判断是否隐藏导航栏的代码改成：

```
setContentView(R.layout.activity_main_screen);
switch (data){
    case "RegisterSuccess":
        Toast.makeText( context: this, text: "欢迎您成为SCOS新客户",Toast.LENGTH_SHORT).show();
        gridShow();
        break;
    case "FromEntry" :
        gridShow();
    case "LoginSuccess":
        gridShow();
    default:
        break;
    }
}
```

4. 使用Intent传递对象

使用Intent 来传递对象通常有两种实现方式，Serializable 和Parcelable，本次我们使用了第一种：

Serializable 是序列化的意思，表示将一个对象转换成可存储或可传输的状态。序列化后的对象可以在网络上进行传输，也可以存储到本地。至于序列化的方法也很简单，只需要让一个类去实现Serializable 这个接口就可以了。

为了传递User对象，我们使得User类实现Serializable接口，如下：

```
public class User implements Serializable{
```

LoginOrRegister活动传递user对象给MainScreen活动：

```
Intent intent = new Intent( packageContext: LoginOrRegister.this,MainScreen.class);
intent.putExtra( name: "extra_data", value: "RegisterSuccess");
// 使User类序列化
intent.putExtra( name: "current_user",loginUser);
startActivity(intent);
```

4. Tablayout + viewpaper + fragment 完成订单页面

先在FoodView活动的布局中加入tablayout 和 Viewpapger布局，整体呈线性排列，布局如下：

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

    <android.support.design.widget.TabLayout
        android:id="@+id/tabLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"></android.support.design.widget.TabLayout>

    <android.support.v4.view.ViewPager
        android:id="@+id/viewPager"
        android:layout_width="match_parent"
        android:layout_height="match_parent"></android.support.v4.view.ViewPager>

</LinearLayout>
```

建立四个Fragment : HotFoodFragment, ColdFoodFragment, SeaFoodFragment, DrinkFoodFragment 均继承自 BaseFoodFragment.

在FoodView中建立个内部类，MyAdapter继承自FragmentPagerAdapter：

```
class MyAdapter extends FragmentPagerAdapter{

    public MyAdapter(FragmentManager fm) { super(fm); }

    @Override
    public Fragment getItem(int i) { return list.get(i); }

    @Override
    public int getCount() {
        return list.size();

    }

    // 设置没个Tab的标题
    @Nullable
    @Override
    public CharSequence getPageTitle(int position) { return titles[position]; }
}
```

将实例化布局，viewPager设定起适配器，再绑定viewpager与tabLayout：

```
// 实例化
viewPager = (ViewPager) findViewById(R.id.viewPager);
tabLayout = (TabLayout) findViewById(R.id.tabLayout);
// 页面，数据源
list = new ArrayList<>();
list.add(new ColdFoodFragment());
list.add(new HotFoodFragment());
list.add(new SeaFoodFragment());
list.add(new DrinkFoodFragment());
// ViewPager的适配器
myadapter = new MyAdapter(getSupportFragmentManager());
viewPager.setAdapter(myadapter);
// 绑定
tabLayout.setupWithViewPager(viewPager);
```

完成导航项的设置，在开始将Fragment加载到PagerView中，在BaseFragment的布局中加入RecycleView

```
<android.support.v7.widget.RecyclerView
    android:id="@+id/recycler_food"
    android:layout_width="match_parent"
    android:layout_height="match_parent"></android.support.v7.widget.RecyclerView>
```

建立Food对象：

```
public class Food {
    private int imageView;
    private String price;
    private String name;
    private boolean order;
    private int store;
```

创建类，FoodAdapter

```
public class FoodAdapter extends RecyclerView.Adapter<FoodAdapter.ViewHolder>{
```

在BaseFoodFragment中将RecycleView绑定适配器：

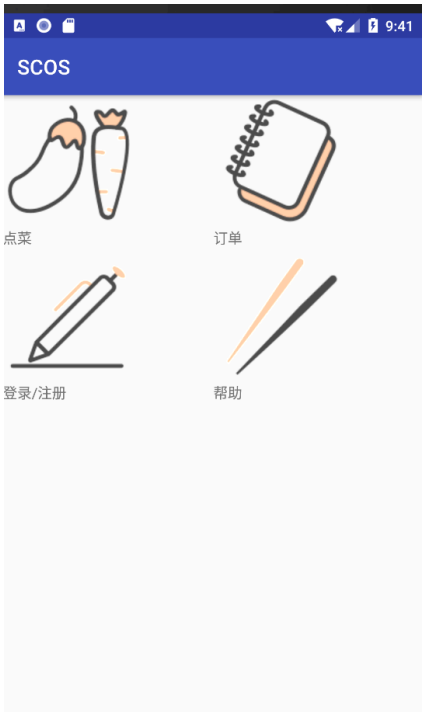
```
RecyclerView recyclerView = view.findViewById(R.id.recycler_food);
FoodAdapter foodAdapter = new FoodAdapter(getActivity(),foodlist);
recyclerView.setLayoutManager(new LinearLayoutManager(getActivity(),LinearLayoutManager.VERTICAL, reverseLayout: false));
recyclerView.setItemAnimator(new DefaultItemAnimator());
recyclerView.setAdapter(foodAdapter);
recyclerView.addItemDecoration(new DividerItemDecoration(getActivity(),DividerItemDecoration.VERTICAL));
return view;
```

完成效果如下：


9:39
SCOS
凉菜
热菜
海鲜
酒水
10元
点菜
蔬菜
20元
点菜
筷子
10元
点菜
蔬菜
20元
点菜


9:39
SCOS
凉菜
热菜
海鲜
酒水
10元
点菜
蔬菜
20元
点菜
筷子
10元
点菜
蔬菜
you clicked view蔬菜
20元
点菜


9:40
SCOS
凉菜
热菜
海鲜
酒水
10元
点菜
蔬菜
20元
退点
筷子
10元
点菜
蔬菜
点菜成功
20元
点菜


9:41
SCOS
点菜
订单
登录/注册
帮助

4. 结论

此次E3作业的工程量比较大，我们对应用也添加了许多业务功能和业务逻辑，基本模拟了下从登录到下单到结账的网上订餐过程。通过这次作业，让我收获许多：

1；学会使用GridView重构导航界面

2；学会使用Intent传递对象，并且进一步熟悉Intent传递信息给下一个活动

3；学会了结合tablayout 结合 Viewpager和Fragment实现一个初步令人满意的订餐界面，实现左右滑动切换Fragment，和上下滑动浏览菜品

当然，本次项目也有许多不足之处，比如UI方面还比较粗糙，没能跟好的使其显示；再次代码方面还比较重复没有更好的复用，比较臃肿。这都是以后需要慢慢完善的。

5. 参考文献

1. https://www.xuebuyuan.com/3218102.html gridView
2. https://www.cnblogs.com/joahyau/p/6549598.html gridView监听
3. https://blog.csdn.net/harrain/article/details/53858079intent 传递对象的两种方式
4. https://blog.csdn.net/bskfnvjtlyzmv867/article/details/70766639 Android 框架之路——Tablayout+ViewPager+Fragment的使用
5. https://blog.csdn.net/qq_33425116/article/details/52599818 Android 导航条效果实现