

软件设计模式

E 1 COS 用例分析

组号：CH

组员：陈相，黄志远

组长：陈相

报告撰写人：黄志远

报告撰写时间：2016/06/18

组员分工

学号	姓名	角色/分工	完成度
SA15226501	黄志远	组员/	100%

1.主题概述

1、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 dashboard 功能导航样式。要求：

- 1) GridView 中导航项为：点菜，查看订单，登录/注册，系统帮助
- 2) 每个导航项都可点击
- 3) 每个导航项标注有图标和文字说明，图标自行设计，采用 png 图片格式

4) 当用户点击“登录/注册”选项时，保持 E 2 原有功能实现

2. 在 SCOS 工程下新建 package 为：es.source.code.model

3. 在包 es.source.code.model 中新建一个 model 类 User.java，User 包含三个域为：userName，password，oldUser，数据类型分别为：String, String, Boolean，每个

5) 域具有 Setter 和 Getter 方法

4. 修改 E 2 中 LoginOrRegister 代码，完成如下功能：

1) 在 LoginOrRegister 界面中添加新按钮“注册”

2) 当用户在 LoginOrRegister 界面点击登录按钮时，按照 E 2 验证规则验证用户输入是否正确。如果正确，则新建 User 对象 loginUser，将用户输入框输入的用户名和密码值通过 Setter 方法赋值到 loginUser 中，并对其 oldUser 域赋值为 True，屏幕跳转至 MainScreen，并向 MainScreen 类传递数据 String 值为“LoginSuccess”，同时将对象 loginUser 传递到 MainScreen 中。如果验证错误，则按 E 2 中要求进行错误提示

3) 当用户在 LoginOrRegister 界面点击注册按钮时，按照 E 2 验证规则验证用户输入是否正确。如果正确，则新建 User 对象 loginUser，将用户输入框输入的用户名和密码值通过 Setter 方法赋值到 loginUser 中，并对其 oldUser 域赋值为 False，屏幕跳转至 MainScreen，并向 MainScreen 类传递数据 String 值为“RegisterSuccess”，同时将对象 loginUser 传递到 MainScreen 中。如果验证错误，则按 E 2 中要求进行错误提示

6) 4) 当用户点击返回按钮时，保持 E 2 中的功能不变

5. 修改 MainScreen.java 代码：

1) 在 MainScreen 中创建一个 User 对象变量 user

2) 判断由 LoginOrRegister 传回的 String 数据值。如果返回数据为“LoginSuccess”，则保持 E 2 功能不变，并将 user 赋值为 LoginOrRegister 传入的 loginUser

3) 如果由 LoginOrRegister 传回的 String 数据值为“RegisterSuccess”，则使用 Toast 提示用户“欢迎您成为 SCOS 新用户”。同时检查“点菜”和“查看订单”导航项是否为隐藏状态，如果为隐藏，则设为显示，并将 user 赋值为 LoginOrRegister 传入的 loginUser

7) 4) 当传入 MainScreen 的 String 值为其他时，将 user 赋值为 NULL

6. 在 SCOS 的 es.source.code.activity 包下新建 Activity 类名为：FoodView，为该

FoodView 定制布局 food_view.xml，要求如下：

- 1) 在布局中使用 ViewPager 显示“冷菜”、“热菜”、“海鲜”、“酒水”四个导航目录
- 2) 支持屏幕左/右滑动导航
- 3) 导航至相应选项时，使用 ListView 加载当前类别的食物列表，并显示到当前屏幕。要求 ListView 每个选项含有：菜名，价格，点菜按钮。当点击点菜按钮时，使用 Toast 提示用户点菜成功，并将按钮显示为“退点”。当点击选项其他区域时，进入到菜品详细页面
- 4) 使用 ActionBar 为 FoodView 添加选项菜单，菜单项为“已点菜品”、
- 8) “查看订单”、“呼叫服务”

7. 在 es.source.code.activity 包下新建 Activity 类名为： FoodOrderView ， 为该 FoodOrderView 定制布局 food_order_view.xml，要求如下：

- 1) 在该页面上使用 ViewPager 显示两个选项卡，分别为“未下单菜”和“已下单菜”
- 2) 支持屏幕左/右滑动导航，和导航项点击导航
- 2) 当用户导航至“已下单菜”，在 FoodOrderView 页面上方使用 ListView 显示订单中的菜品项，每项中显示：菜名，价格，数量，备注；在页面下方显示菜品总数、订单总价及“结账”按钮
- 3) 当用户导航至“未下单菜”，在 FoodOrderView 页面上方使用 ListView 显示已点但未生成订单菜品，每项显示：菜名，价格，数量，备注，“退点”按钮；在页面下方显示菜品总数、订单总价及“提交订单”按钮

8. 在 es.source.code.activity 包下新建 Activity 类名为： FoodDetailed ， 为该 FoodDetailed 定制布局 food_detailed.xml，要求如下：

- 1) 在该布局中显示菜品信息：菜品图片，菜名，菜价，菜品备注输入框
- 2) 在该布局中添加一个按钮，按钮标签根据当前菜品是否已在已点菜品中，显示不同标签：如果是已点菜，则按钮显示“退点”，否则显示“点菜”
- 3) FoodDetailed 支持左/右滑动切换菜品。当用户在当前屏幕左/右滑动时，当前屏幕显示不同的菜品详细信息

9. 修改 FoodOrderView、FoodView 代码，在其中添加 User 变量 user

10. 修改 MainScreen 代码，完成如下功能：

- 1) 当用户点击“点菜”时，页面跳转至 FoodView，并使用 Intent 将 user 作为数据传给 FoodView 的 user 变量
- 2) 当用户点击“查看订单”时，页面跳转至 FoodOrderView，并使用 Intent
- 10) 将 user 作为数据传给 FoodOrderView 的 user 变量

11. 修改 FoodView 代码，当用户点击食物列表项时，页面跳转至 FoodDetailed；当用户点击 ActionBar “已点菜品”时，页面跳转至 FoodOrderView，并默认显示为“未下单菜”选项页面，传入 user 变量；当用户点击 ActionBar “查看订单”时，页面跳转至 FoodOrderView，并默认显示为“已下单菜”选项页面，传入 user 变量

12. 修改 FoodOrderView 代码，当用户点击“结账”按钮时，当 user 不为 null 时，判断 user 中的 oldUser 值；如果为 True，则使用 Toast 提示“您好，老顾客，本次你可享受 7 折优惠”；否则，不作提示

13. 请根据以上任务，自主添加所需数据结构和功能代码，适当扩展功能实现，完成 SCOS 的基本原型开发，并调试正确运行
14. 修改 SCOS 工程的 `AndroidManifest.xml` 中 `versionCode` 和 `versionName` 为 2.0，
11) 打包发布 SCOS2.0.apk，并用真机测试运行

2、假设

开发工具：Android studio

Android SDK：19

3、实现或证明

实验过程以及结果如下：

- 1、修改 E 2 中的 MainScreen 布局，形成 dashboard 功能导航样式，当用户点击“登录、注册”选项时，保持 E 2 原有功能实现。

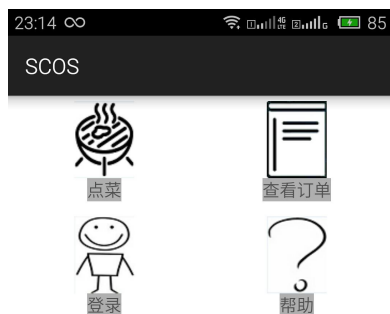


图 1 dashboard 主界面



图 2 登录注册界面

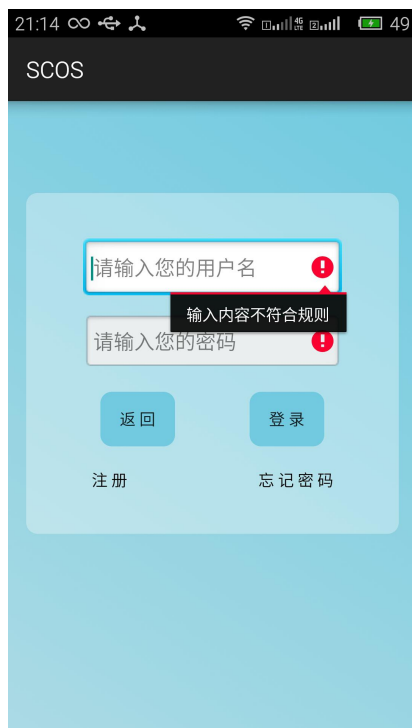


图 3 登录验证界面

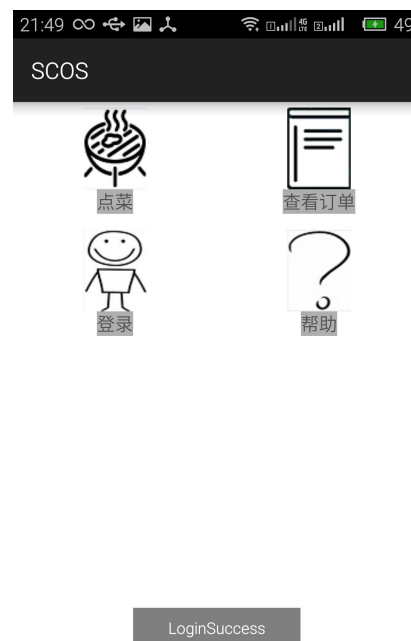


图 4 注册成功界面

2、新建 package 为 es.source.code.model，并新建 model 类 user.java。

```
package es.source.code.model;
import java.io.Serializable;
/**
 * Createa by huang on 2016/6/13.
 */
public class User implements Serializable{
    private String userName;
    private String passwrod;
    private Boolean oldUser;
    public User() {}
    public User(String name,String passwrod,Boolean oldUser){
        this.userName = name;
        this.passwrod = passwrod;
        this.oldUser = oldUser;
    }
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getPasswrod() {
        return passwrod;
    }
    public void setPasswrod(String passwrod) {
        this.passwrod = passwrod;
    }
    public Boolean getOldUser() {
        return oldUser;
    }
    public void setOldUser(Boolean oldUser) {
        this.oldUser = oldUser;
    }
    @Override
    public String toString() {
        return "User{" +
            "userName=' " + userName + '\',' +
            ", passwrod=' " + passwrod + '\',' +
            ", oldUser=" + oldUser +
            '}' ;
    }
}
```

3、修改 E 2 中 LoginOrRegister 代码，添加“注册”按钮（见上图图 2），登录成功，新建 User 对象 loginUser,并赋值到 loginUser 中，对其 oldUser 域赋值为 True，跳转至 mainScreen，传递 String 值为 LoginSuccess，同时传递 loginUser 对象；注册时，验证输入是否合法，如果正确新建 User 对象 loginUser,将用户输入框中的用户名和密码赋值到 loginUser 中，并对 oldUser 属性赋值为 False，跳转到 mainScreen，传递 loginUser 对象和 String 值“RegisterSuccess”。

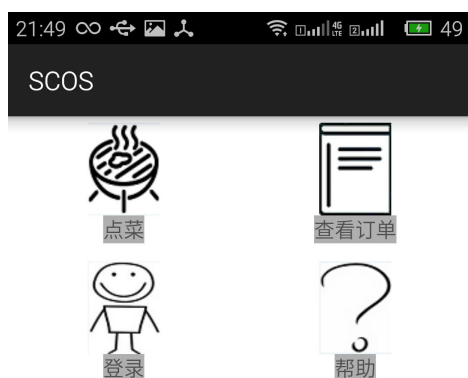


图 5 登录成功界面



图 6 注册成功界面

4、在 es.source.code.activity 包下新建 Activity 类 FoodView，在布局中使用 ViewPager 显示冷菜、热菜、海鲜、酒水四个导航目录；支持屏幕左右滑动；导航至相应选项时，使用 listview 加载该类别的食物列表，并显示到当前屏幕，点击点菜按钮时，提示点菜成功，并将按钮显示为“退点”，点击选项其他区域是进入到菜品详细页面；使用 ActionBar 为 FoodView 添加选项卡菜单。



图 7 冷菜选项卡界面



图 8 热菜选项卡界面



图 9 海鲜选项卡界面



图 10 酒水选项卡界面



图 11 菜品详细界面



图 12 冷菜选菜成功界面

5、在 es.source.code.activity 包下新建 Activity 类名为 FoodOrderView，使用 Viewpager 显示“未下单菜”和“已下单菜”选项卡；

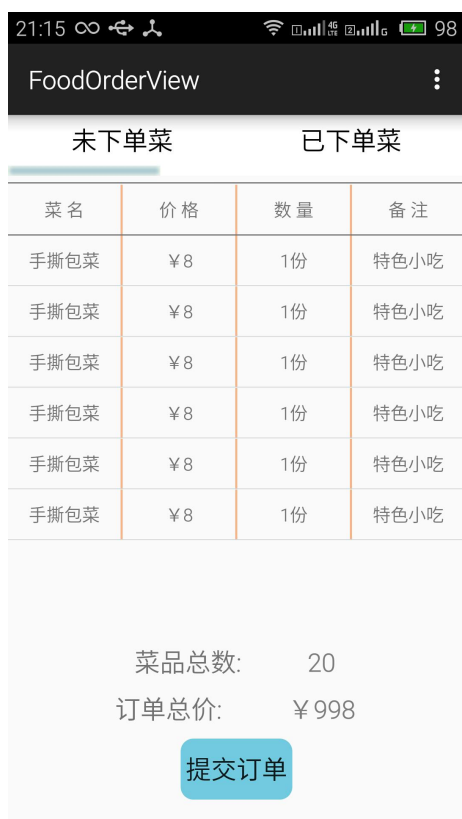


图 13 未下单菜界面

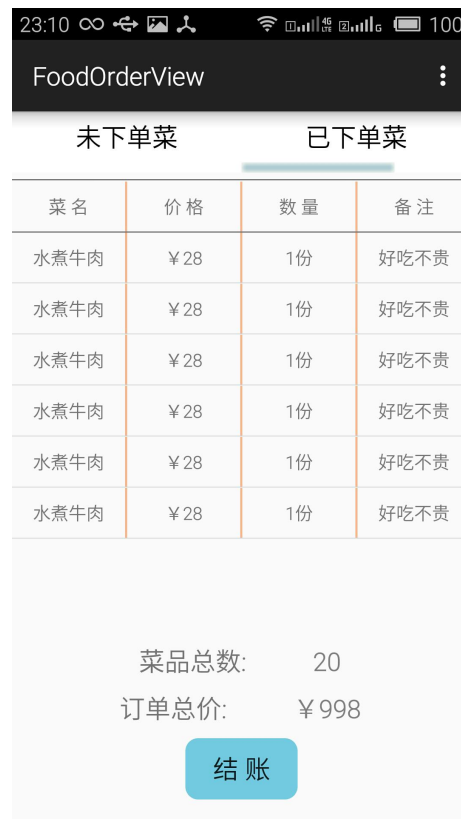


图 14 已下单菜界面

6、在 es.source.code.activity 包下新建 Activity 类名为：FoodDetailed，布局中包含菜品信息，按钮显示退点、点菜状态，FoodView 支持左右滑动切换菜品。



图 15 菜品信息切换效果 1



图 16 菜品信息切换效果 2



图 17 菜品信息切换效果 3



图 18 菜品信息切换效果 4

- 7、修改 FoodOrderView、FoodView 代码，添加 User 变量 user。
- 8、修改 mainScreen 代码，当用户点击“点菜”时，页面跳转到 FoodView 并传递 user 变量，用户点击“查看订单”时，页面跳转到 FoodOrderView，并传递 user 变量。
- 9、修改 FoodView 代码，用户点击食物列表项时，页面跳转至 FoodDetailed（参见图 11），当用户点击 ActionBar “已点菜品”时，页面跳转至 FoodOrderViewb 并默认显示为“为下单菜”选项页面（参见图 13），当用户点击 ActionBar “查看订单”时，页面跳转至 FoodOrderView，并默认显示“以下菜单”选项页面（参见图 14）。
- 10、修改 FoodOrderView 代码，用户点击“结账”按钮时，若 user 不为 null，判断 user 中的 oldUser 是否为 True，若是则提示“您好，老顾客，本次你可享受 7 折优惠”，否则不作提示。

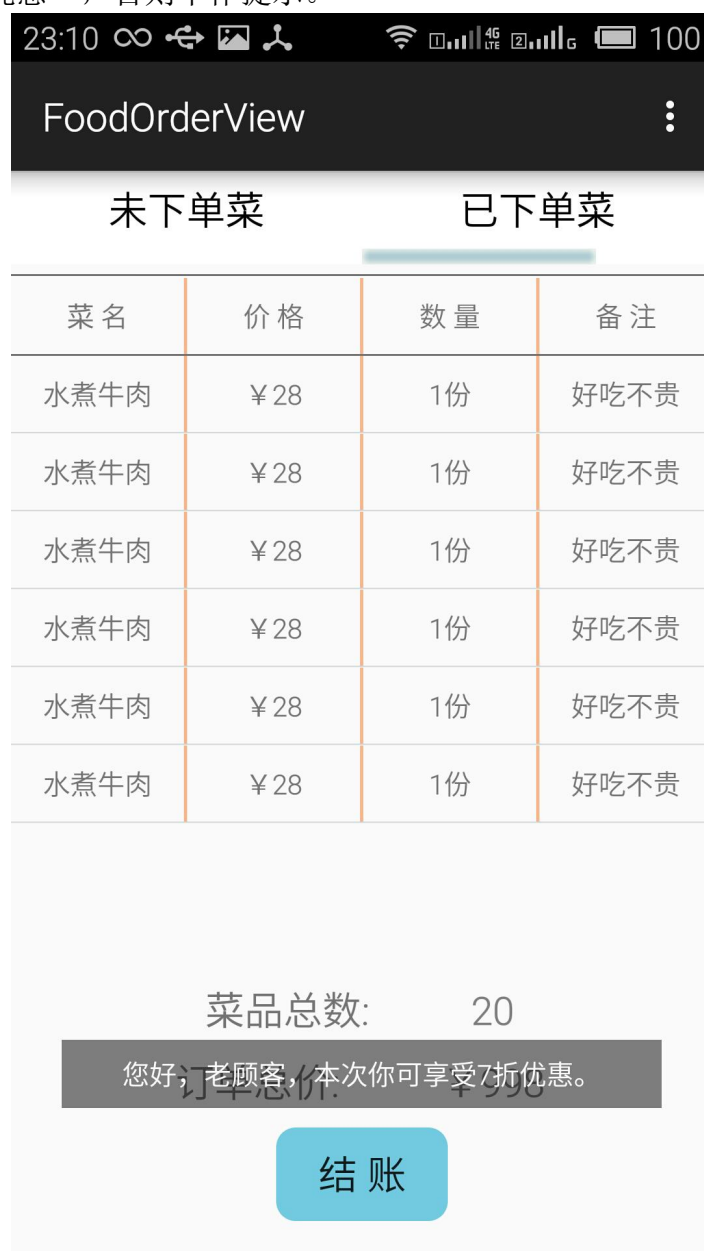


图 19 老顾客结账优惠提示

2. 结论

此次试验的主要内容是练习 Custom Adapter 和 View Component，其中 Adapter 多和 listview 配合使用，试验中界面结合了 viewpager。Adapter 是连接后端数据和前端显示的适配器接口，是数据和 UI（View）之间一个重要的纽带。在常见的 View(ListView,GridView) 等地方都需要用到 Adapter。如下图直观的表达了 Data、Adapter、View 三者的关系：

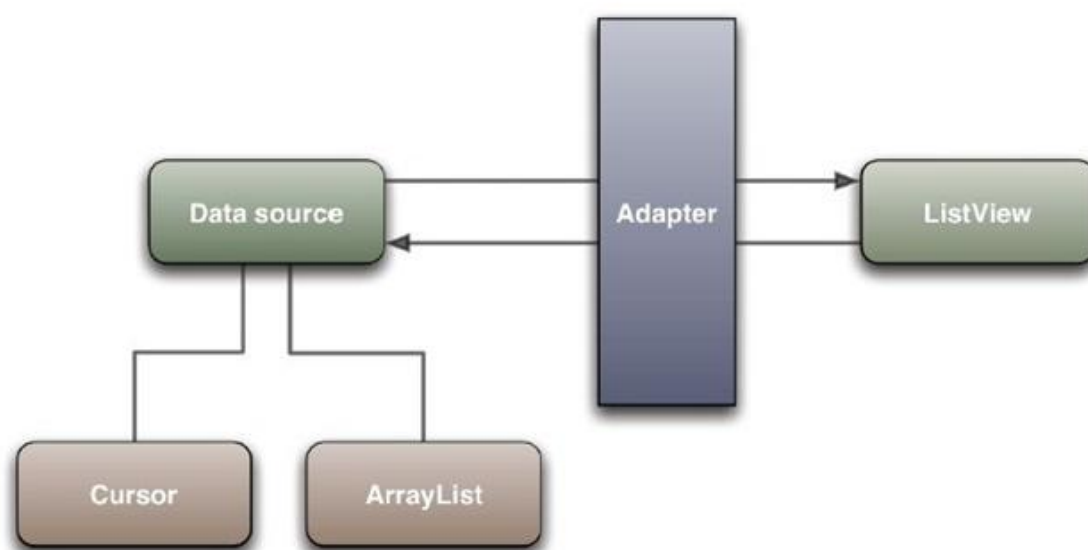


图 20 Data、Adapter、View 三者关系

由图可以看到在 Android 中与 Adapter 有关的所有接口、类的完整层级图。在我们使用过程中可以根据自己的需求实现接口或者继承类进行一定的扩展。比较常用的有 BaseAdapter，SimpleAdapter，ArrayAdapter，SimpleCursorAdapter 等。

- BaseAdapter 是一个抽象类，继承它需要实现较多的方法，所以也就具有较高的灵活性；
- ArrayAdapter 支持泛型操作，最为简单，只能展示一行字。
- SimpleAdapter 有最好的扩充性，可以自定义出各种效果。
- SimpleCursorAdapter 可以适用于简单的纯文字型 ListView，它需要 Cursor 的字段和 UI 的 id 对应起来。如需要实现更复杂的 UI 也可以重写其他方法。可以认为是 SimpleAdapter 对数据库的简单结合，可以方便地把数据库的内容以列表的形式展示出来。

3.参考文献

无

6.会议备注

会议时间：2016/06/18

地点： 思贤楼 209

参加人员：黄志远、陈相

主要内容：关于实验三进行讨论、分工