

中山大学移动信息工程学院本科生实验报告

(2017 年秋季学期)

课程名称 : 移动应用开发

任课教师 : 郑贵锋

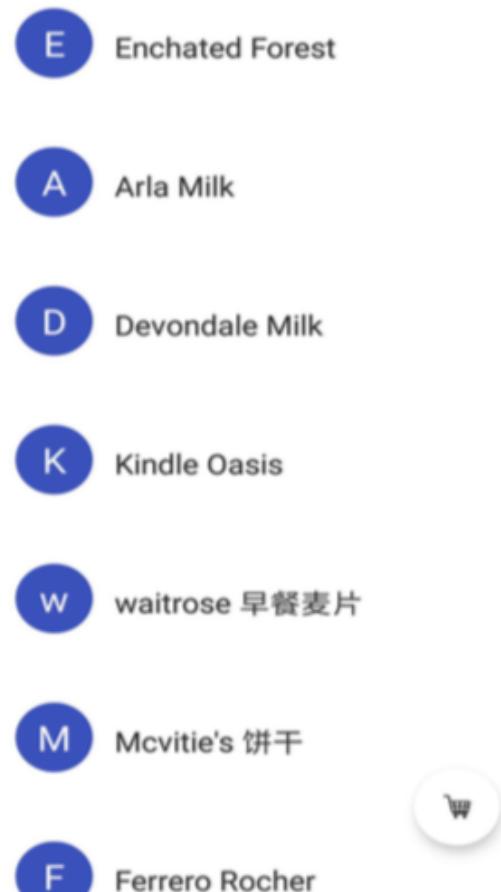
年级	2015	专业 (方向)	移动 (互联网方向)
学号	15352151	姓名	柯博
电话	13671412922	Email	kebo@mail2.sysu.edu.cn
开始日期		完成日期	

一、 实验题目

Intent、Bundle 的使用以及 RecyclerView、ListView 的应用

二、 实现内容

本次实验模拟实现一个商品表，有两个界面，第一个界面用于呈现商品，如下所示：



下面还有数据，就不截图了，数据在素材中有给出。

点击右下方的悬浮按钮可以切换到购物车：



上面两个列表点击任意一项后，可以看到详细信息：



实验要求：布局方面的要求：

1.商品表界面

每一项为一个圆圈和一个名字，圆圈与名字均竖直居中。圆圈中为商品名字的首字母，首字母处于圆圈

的中心，首字母为白色，名字为黑色，圆圈的颜色自定义即可，建议用深色的颜色，否则白色的首字母可能看不清。

2.购物车列表界面

在商品表界面的基础上增加一个价格，价格为黑色。

3.商品详情界面顶部

顶部占整个界面的 1/3.每个商品的图片在商品数据中已给出，图片与这块 view 等高。返回图标处于这块 view 的左上角，商品名字处于左下角，星标处于右下角，它们与边距都有一定距离，自己调节即可。需要注意的是，返回图标与名字左对齐，名字与星标底边对齐。这一块建议大家使用 RelativeLayout 实现，以便熟悉 RelativeLayout 的使用。

4.商品详情界面中部

使用的黑色 argb 编码值为#D5000000，稍微偏灰色一点的“重量”、“300g”的 argb 编码值为 #8A000000。注意，价格那一栏的下边有一条分割线，argb 编码值为#1E000000，右边购物车符号的左边也有一条分割线，argb 编码值也是#1E000000，这条分割线要求高度与聊天符号的高度一致，并且竖直居中。字体的大小看着调就可以。“更多资料”底部的分割线高度自己定，argb 编码与前面的分割线一致。

5.商品详情页面底部

一键下单

分享商品

不感兴趣

查看更多商品促销信息

这个没什么说的，内容和样式都很清楚。

6.特别提醒，这次的两个界面顶部都没有标题栏，要用某些方法把它们去掉。

逻辑方面的要求：

1.使用 RecyclerView 实现商品列表。点击商品列表中的某一个商品会跳转到该商品详情界面，呈现该商品的详细信息；长按商品列表中的第 i 个商品会删除该商品，并弹出 Toast 提示“移除第 i 个商品”

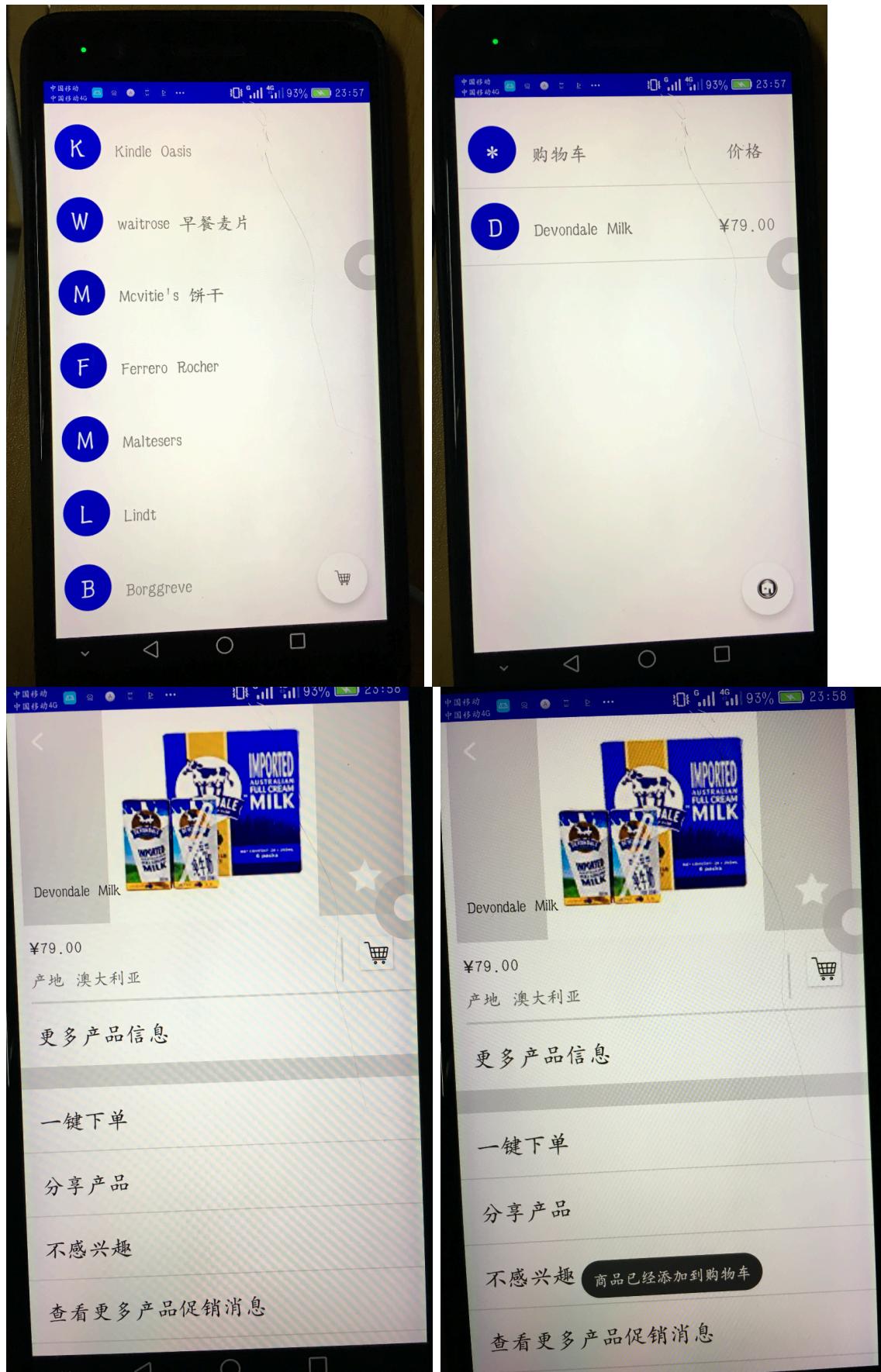
2.点击右下方的 FloatingActionButton，从商品列表切换到购物车或从购物车切换到商品列表，并且 FloatingActionButton 的图片也要做相应改变。可通过设置 RecyclerView 不可见，ListView 可见来实现从商品列表切换到购物车。可通过设置 RecyclerView 可见，ListView 不可见来实现从购物车切换到商品列表。

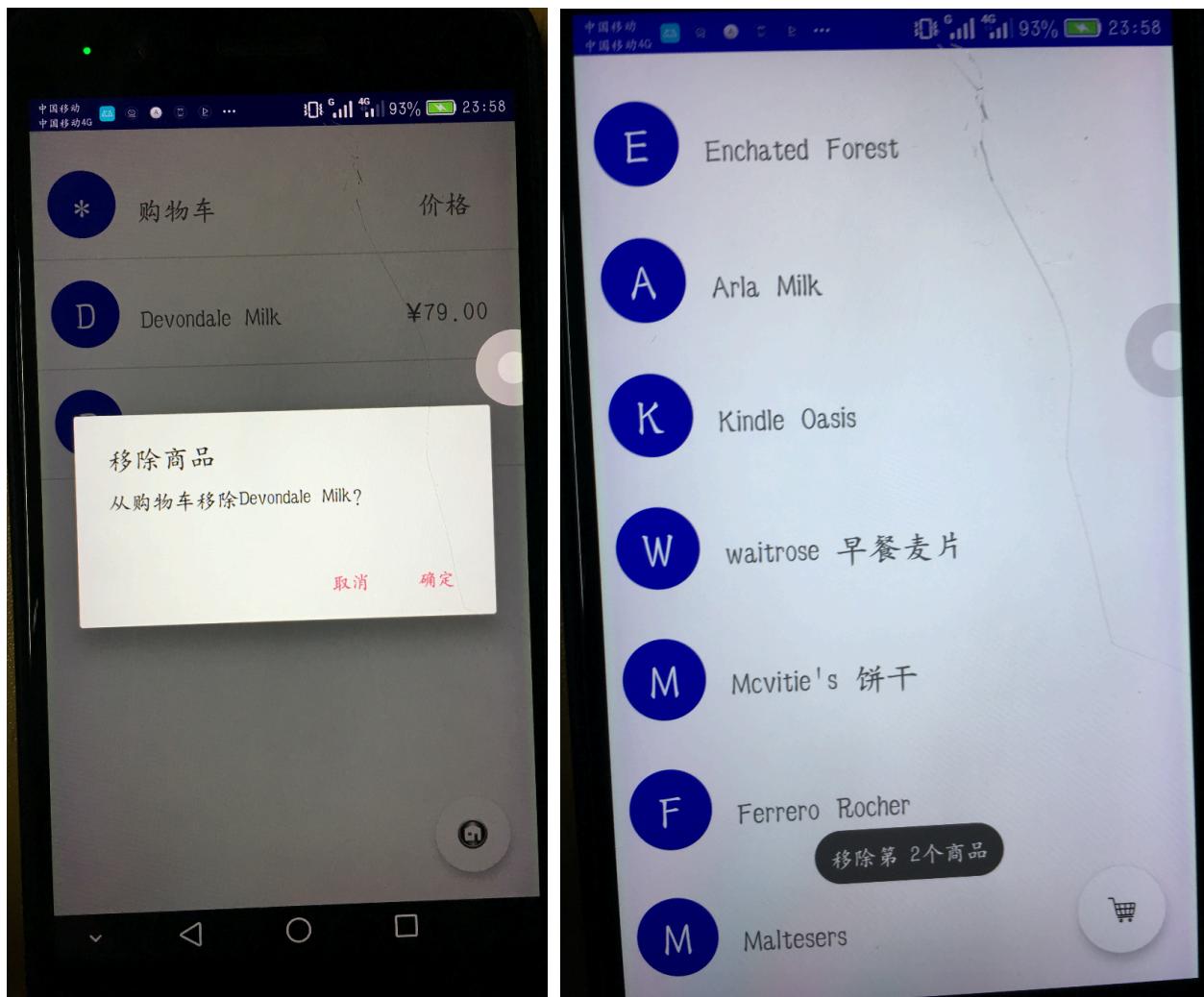
3.使用 ListView 实现购物车。点击购物车的某一个商品会跳转到该商品详情页面，呈现该商品的详情信息；长按购物车中的商品会弹出对话框询问是否移除该商品，点击确定则移除该商品，点击取消则对话框消失。

4.商品详情界面中点击返回图标会返回上一层，点击星标会切换状态，如果原先是空心星星，则会变成实心星星；如果原先是实心星星，则会变成空心星星。点击购物车图标会将该商品添加到购物车中并弹出 Toast 提示。

三、课堂实验结果

(1) 实验截图





(2) 实验步骤以及关键代码

```
import android.app.Application;
//重载android自带的Application类为ShoppingApp，用于全局变量的访问
public class ShoppingApp extends Application {
    public ItemData item_data;
    public ItemData chart_data;
    public MainActivity.MyListViewAdapter listview_adpater;

    @Override
    public void onCreate() {
        super.onCreate();
        item_data = new ItemData( type: true); //商品列表中的商品
        chart_data = new ItemData( type: false); //购物车中的商品
    }
}
```

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    //声明需要用到的控件
    protected RecyclerView rv_items;
    protected ListView lv_items;
    protected ShoppingApp my_app;
    protected FloatingActionButton fab_tochart;
    protected int mode; // mode=1:商品列表; mode=2:购物车
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //初始化activity， 初始化各种变量
    my_app = (ShoppingApp) getApplication();
    rv_items = (RecyclerView) findViewById(R.id.itemlist);
    lv_items = (ListView) findViewById(R.id.chartlist);
    fab_tochart = (FloatingActionButton) findViewById(R.id.to_chart);
    mode = 1;

    lv_items.setVisibility(View.GONE); //购物车不显示
    //设置FloatingActionButton的点击事件
    fab_tochart.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(mode == 1) mode = 2;
            else mode = 1;
            if(mode == 1){ //切换显示的页面
                lv_items.setVisibility(View.GONE);
                rv_items.setVisibility(View.VISIBLE);
                fab_tochart.setImageResource(R.mipmap.shoplist);
            } else {
                lv_items.setVisibility(View.VISIBLE);
                rv_items.setVisibility(View.GONE);
                fab_tochart.setImageResource(R.mipmap.mainpage);
            }
        }
    });
}

// Item Part.
MyRcAdpater rcad = new MyRcAdpater( ctx: MainActivity.this, R.layout.view_item,
    new ArrayList<>(my_app.item_data.data), new ArrayList<>(my_app.item_data.itemid)) {
    @Override //用adapter填充listview
    public void convert(MyRcViewHolder vh, Map<String, Object> m) {
        TextView icon = vh.getView(R.id.item_icon);
        TextView name = vh.getView(R.id.item_name);
        TextView price = vh.getView(R.id.item_price);
        name.setText(m.get("name").toString());
        icon.setText(m.get("firstletter").toString());
        price.setText("");
    }
};

//设置商品列表Item的点击属性
rcad.setOnItemClickListener(new MyOnItemClickListener() {
    @Override //单击进入商品详情
    public void onClick(MyRcAdpater rcad, int position) {
        Intent it_item = new Intent();
        it_item.setClass( packageContext: MainActivity.this, DetailActivity.class);
        it_item.putExtra( name: "itemid", rcad.ad_dataid.get(position));
        startActivity(it_item);
    }

    @Override //长按移除商品
    public void onLongClick(MyRcAdpater rcad, int position) {
        rcad.ad_dataid.remove(position);
        rcad.ad_data.remove(position);
        rcad.notifyItemRemoved(position);
        rcad.notifyItemRangeChanged(position, rcad.ad_data.size());
        Toast.makeText(rcad.ad_ctx,
            text: "移除第 " + position + "个商品", Toast.LENGTH_SHORT).show();
    }
});

```

```

// Chart Part.
final MyListViewAdapter lvad = new MyListViewAdapter( ctx: MainActivity.this,
    my_app.chart_data.data, my_app.chart_data.itemid);
my_app.listview_adpater = lvad;
lv_items.setAdapter(lvad);
//设置购物车中item的单击属性
lv_items.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        if(position != 0) {
            Intent it_item = new Intent();
            it_item.setClass( packageContext: MainActivity.this, DetailActivity.class);
            it_item.putExtra( name: "itemid", my_app.chart_data.itemid.get(position));
            startActivity(it_item);
        }
    }
});
//设置长按属性
lv_items.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> parent, View view,
                                   final int position, long id) {
        //弹出对话框
        if(position != 0) {
            AlertDialog.Builder ad_builder;
            ad_builder = new AlertDialog.Builder( context: MainActivity.this);
            ad_builder
                .setTitle("移除商品")
                .setMessage("从购物车移除"
                    + my_app.chart_data.data.get(position).get("name").toString()
                    + "? ")
                .setNegativeButton( text: "取消", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int i) {
                })
                .setPositiveButton( text: "确定", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int i) {
                        my_app.chart_data.data.remove(position);
                        my_app.chart_data.itemid.remove(position);
                        my_app.chart_data.stared.remove(position);
                        lvad.notifyDataSetChanged();
                    }
                })
                .create().show();
        }
        return true;
    }
});
public class MyRcAdpater extends RecyclerView.Adapter<MyRcAdpater.MyRcViewHolder> {
//自定义RecyclerView Adapter
    ArrayList<Map<String, Object>> ad_data;
    ArrayList<Integer> ad_dataid;
    Context ad_ctx;
    int ad_lid;
    MyOnItemClickListener ad_click;
    //重载构造函数
    public MyRcAdpater(Context ctx, int lay_id, ArrayList<Map<String, Object>> data,
                        ArrayList<Integer> data_id) {
        ad_ctx = ctx;
        ad_lid = lay_id;
        ad_data = data;
        ad_dataid = data_id;
    }
}

```

```
//自定义ListViewAdapter
public class MyListViewAdapter extends BaseAdapter {
    Context lv_ctx;
    ArrayList<Map<String, Object>> lv_data;
    ArrayList<Integer> lv_dataid;

    public MyListViewAdapter(Context ctx, ArrayList<Map<String, Object>> data,
                           ArrayList<Integer> data_id) {
        lv_ctx = ctx;
        lv_data = data;
        lv_dataid = data_id;
    }
}
```

ItemData.java

```
public ItemData(boolean type) { //ItemData构造函数
    data = new ArrayList<>(); //初始化
    stared = new ArrayList<>();
    itemid = new ArrayList<>();
    imgid = new ArrayList<>();
    if (type) { //根据参数确定生成ItemData的类型
        for (int i = 0; i < name.length; ++i) {
            stared.add(0); //默认收藏列表全为False
            itemid.add(i);
            imgid.add(resimg[i]);
            Map<String, Object> tmp = new LinkedHashMap<>(); //一个中间Map变量temp
            tmp.put("name", name[i]);
            tmp.put("price", price[i]);
            tmp.put("infotype", infotype[i]);
            tmp.put("infovalue", infovalue[i]);
            tmp.put("firstletter", firstletter[i]);
            data.add(tmp);
        }
    } else {
        Map<String, Object> tmp = new LinkedHashMap<>();
        stared.add(-1);
        itemid.add(-1);
        imgid.add(-1);
        tmp.put("name", "购物车");
        tmp.put("price", "价格");
        tmp.put("infotype", "title");
        tmp.put("infovalue", "title");
        tmp.put("firstletter", "*");
        data.add(tmp);
    }
}
```

DetailActivity.java

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState); //初始化商品详情这个activity
    setContentView(R.layout.activity_detail); //从xml文件加载布局

    my_app = (ShoppingApp) getApplication(); //读取全局变量
    int iid = getIntent().getIntExtra("name: "itemid", defaultValue: 0);
    //Intent传递的内容
    idx = my_app.item_data.getIndex(iid);
    //findViewById()方法获取xml布局中的元素
    t_price = (TextView) findViewById(R.id.price);
    t_name = (TextView) findViewById(R.id.name);
    t_type = (TextView) findViewById(R.id.type);
    t_info = (TextView) findViewById(R.id.info);
    b_addchart = (Button) findViewById(R.id.addchart);
    b_back = (Button) findViewById(R.id.back);
    b_star = (Button) findViewById(R.id.star);
    i_img = (ImageView) findViewById(R.id.itemimg);
    lv_more = (ListView) findViewById(R.id.more);
    lv_op = (ListView) findViewById(R.id.operation);
    //获取ItemData商品详情
    t_price.setText(my_app.item_data.data.get(idx).get("price").toString());
    t_name.setText(my_app.item_data.data.get(idx).get("name").toString());
    t_type.setText(my_app.item_data.data.get(idx).get("infotype").toString());
    t_info.setText(my_app.item_data.data.get(idx).get("infovalue").toString());
    i_img.setImageResource(my_app.item_data.imgid.get(idx));

    b_back.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { //用finish()方法设置返回按钮
            finish();
        }
    });
    update_star(); //更新收藏星星状态
    b_star.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { //设置点击收藏按钮事件
            int val = my_app.item_data.stared.get(idx);
            if (val == 1) val = 0;
            else val = 1;
            my_app.item_data.stared.set(idx, val);
            update_star();
        }
    });
    b_addchart.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { //设置添加购物车按钮事件
            my_app.chart_data.add( //将当前商品加入购物车的列表中
                my_app.item_data.data.get(idx),
                my_app.item_data.stared.get(idx),
                my_app.item_data.itemid.get(idx),
                my_app.item_data.imgid.get(idx)
            );
            my_app.listview_adpater.notifyDataSetChanged(); //刷新列表
            Toast.makeText(context: DetailActivity.this,
                           text: "商品已经添加到购物车", Toast.LENGTH_SHORT).show();
        }
    });
    final String[] s_more = {"更多产品信息"}; //商品详情页的操作按钮，以Array形式实现
    final String[] s_op = {"一键下单", "分享产品", "不感兴趣", "查看更多产品促销消息"};
    lv_more.setAdapter(new ArrayAdapter<>( context: DetailActivity.this, R.layout.just_text, s_more));
    lv_op.setAdapter(new ArrayAdapter<>( context: DetailActivity.this, R.layout.just_text, s_op));
}

public void update_star() { //更新星星的状态
    if (my_app.item_data.stared.get(idx) == 0) { //查看收藏列表中对应位置的值
        b_star.setBackgroundResource(R.mipmap.empty_star);
    } else {
        b_star.setBackgroundResource(R.mipmap.full_star);
    }
}

```

(3) 实验遇到困难以及解决思路

实验遇到的困难：一开始什么都不懂，看着 PDF 好几天都是一脸懵逼，我在哪？我要干嘛？这在写什么？想上个 GitHub 还偏偏赶上特殊时期不能上网，这就很皮。看了几遍 PDF 之后总算是明白了实验文档在写什么和我应该要学什么，但还是不知道怎么用到代码中。参考借鉴了一下往年师兄师姐的报告之后恍然大悟，要先搭建起 class 的框架，之后再实现各种各样的内容，要时刻牢记 Java 是一门面向对象的语言，所有的东西都是对象，所有行动都需要通过函数调用，与 C++、python 等面向过程等语言要区分开来，一定要先写 class 框架。借助于最新的 Android Studio 3.0，我在编写接口（如 Adapter）和其他函数的时候，自动补全功能帮助良多，没有发生什么错误，其实 PDF 上各种方法的例子都写的比较清楚，TA 们也很认真地解答我们的疑问，加上善用百度，很多错误可以在课堂上解决完毕。

四、课后实验结果

五、实验思考及感想

第一次写这么多的 Java 代码……从 XML 界面设计到 Java 事件处理，我们对于 Android 应用开发的学习已经由浅入深，开始接触核心的用户交互层面。按耐不住内心的激动，很想沉迷学习，对期中和期末的 project 跃跃欲试了呢！

作业要求：

1. 命名要求：学号_姓名_实验编号，例如 15330000_林 XX_lab1。
2. 实验报告提交格式为 pdf。
3. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按 0 分处理。