

中山大学数据科学与计算机学院本科生实验报告

(2017 年秋季学期)

课程名称：手机应用平台开发

任课教师：刘宁

年级	15 级软件工程	专业 (方向)	计算机应用
学号	15331236	姓名	马朝露
电话	15521122675	Email	1073627941@qq.com
开始日期	2017/10/21	完成日期	2017/10/23

一、 实验题目

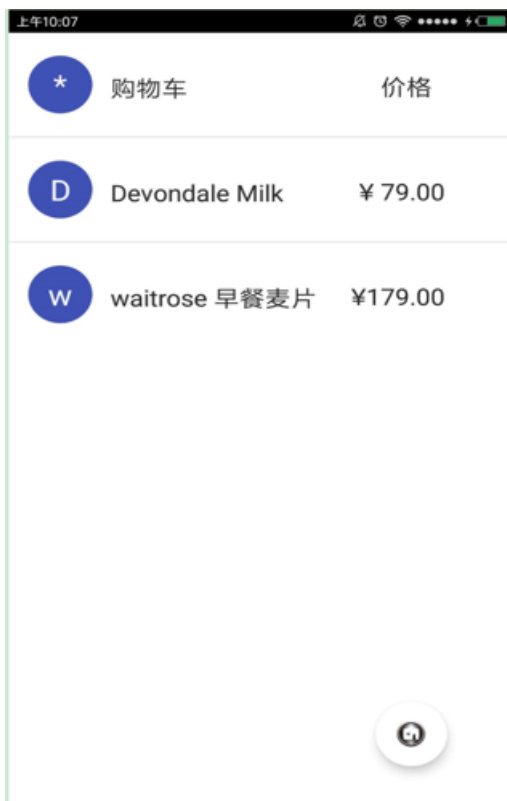
Intent、Bundle 的使用以及 RecyclerView、ListView 的应用

二、 实现内容

【实验内容】本次试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



点击右下方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，可以看到详细的信息：



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名字的首字母，首字母要处于圆圈的中心，首字母为白色，名字为黑色，圆圈的颜色自定义。

2.购物车列表界面

在商品表的界面基础上增加一个价格，价格为黑色

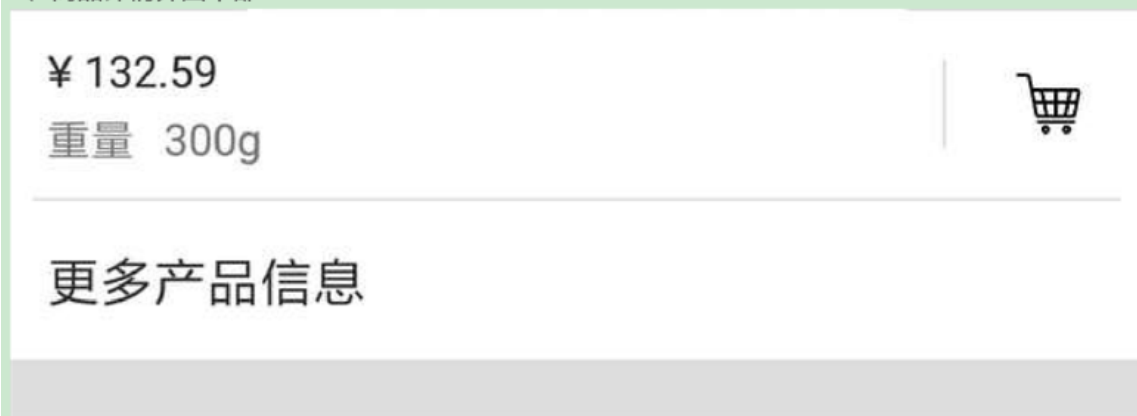
3.商品详情界面顶部



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,自己调出合适的距离即可。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

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了。“更多资料”底部的分割线高度自己定, argb编码值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

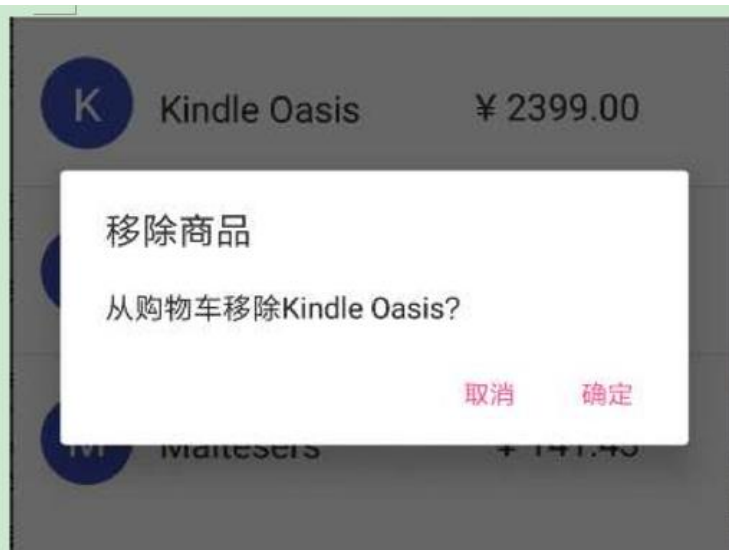
查看更多商品促销信息

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，并且弹出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商品详情界面，呈现该商品的详细信息；长按购物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。



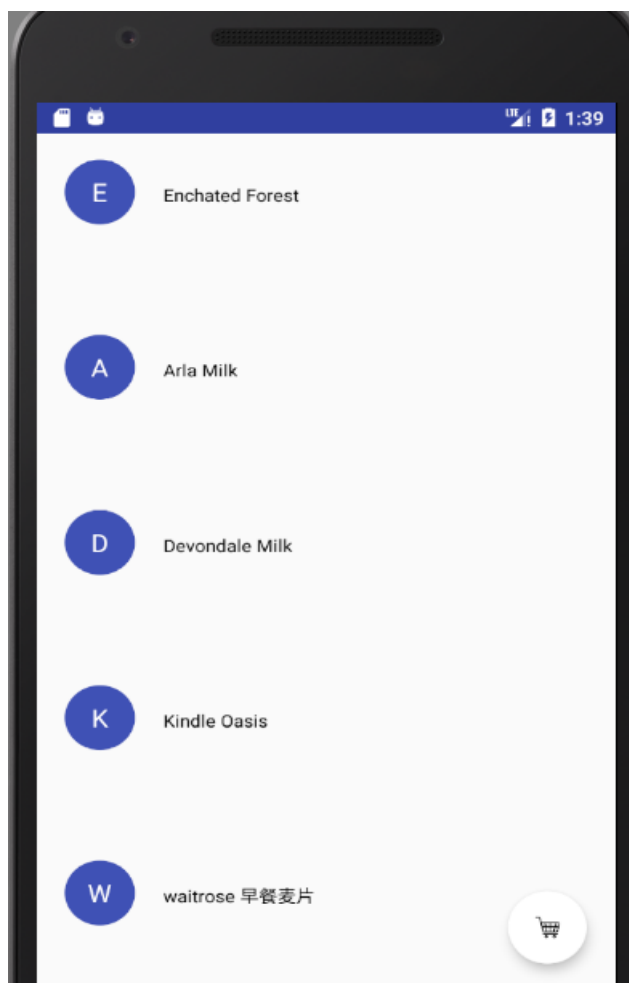
注意对话框中的商品名为被长按的商品。

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:"商品已添加到购物车"。

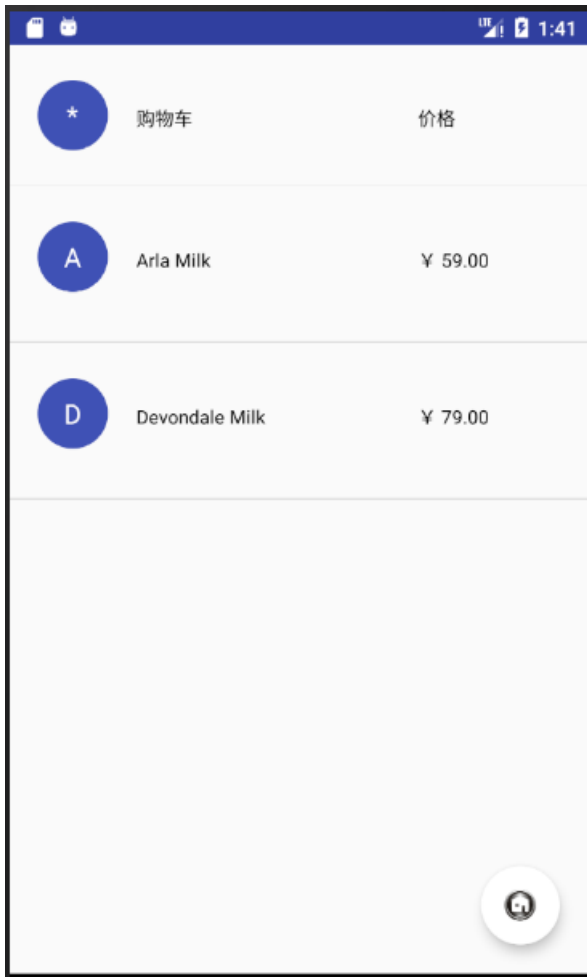
注:不要求判断购物车是否已有该商品,即如果已有一件该商品,添加之后则显示两个即可。未退出商品详细界面时,点击多次购物车图标可以只添加一件商品也可以添加多件到购物车中。

三、 课堂实验结果

(1) 实验截图



商品表界面



购物车界面



商品详情页面



从购物车移除商品

(2) 实验步骤以及关键代码

1.商品表界面（只截取作业文档中没有给出详细提示的部分）

```

public class ViewHolder extends RecyclerView.ViewHolder {
    private SparseArray<View> mViews; // 存储list_Item的子View
    private View mConvertView; // 存储List_Item
    public ViewHolder(Context context, View itemView, ViewGroup parent) {
        super(itemView);
        mConvertView = itemView;
        mViews = new SparseArray<View>();
    }

    public static ViewHolder get(Context context, ViewGroup parent, int layoutId) {
        View itemView = LayoutInflater.from(context).inflate(layoutId, parent, false);
        ViewHolder holder = new ViewHolder(context, itemView, parent);
        return holder;
    }

    public <T extends View> T getView(int ViewId) {
        View view = mViews.get(ViewId);
        if (view == null) {
            //创建View
            view = mConvertView.findViewById(ViewId);
            // 将View存入mViews
            mViews.put(ViewId, view);
        }
        return (T)view;
    }
}

```

ViewHolder的实现，ViewHolder相当于一个存放每一Item的View的容器。

```

public void onBindViewHolder(final ViewHolder holder, int position) {
    Map<String, Object> s = mDatas.get(position);
    TextView name = holder.getView(R.id.name);
    name.setText(s.get("name").toString());
    TextView initial = holder.getView(R.id.initial);
    initial.setText(s.get("initial").toString());
    if(mOnItemClickListener != null){
        //为ItemView设置监听器
        holder.itemView.setOnClickListener((v) -> {
            int position = holder.getLayoutPosition(); // 1
            mOnItemClickListener.onClick(holder.itemView, position); // 2
        });
    }
    if(mOnItemLongClickListener != null){
        holder.itemView.setOnLongClickListener((v) -> {
            int position = holder.getLayoutPosition();
            mOnItemLongClickListener.onLongClick(holder.itemView, position);
            //返回 true 表示消耗了事件 事件不会继续传递
            return true;
        });
    }
}
}

```

MyRecyclerView中的onBindViewHolder函数，在这个函数中实现item点击事件函数接口。其中点击事件前几行的代码相当于完成一个convert。将xml中的item绑定ViewHolder中的数据，将数据显示在对应的textView上

悬浮按钮的实现：

```

<android.support.design.widget.FloatingActionButton
    android:id="@+id/FloatingActionButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/shoplist"
    app:backgroundTint="#FFFFFF"
    app:backgroundTintMode="src_atop"
    app:rippleColor="#FFFFFF"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_margin="20dp"
/>

```

2.购物车界面

```

<ListView
    android:layout_below="@id/bb"
    android:id="@+id/list"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</ListView>

```

```

this.adapter = new SimpleAdapter(this, mDataas_, R.layout.shoplist_item,
    new String[] {"initial_1", "name_1", "price"},
    new int[] {R.id.initial_1, R.id.name_1, R.id.price});

```

购物车我使用ListView来实现

3. 商品详情界面

```

<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:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorGray"
    android:weightSum="3"
    tools:context="com.example.zhaoluma.lab3.details">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="2dp"
        android:layout_weight="1">

        <ImageButton
            android:id="@+id/back"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentTop="true"
            android:layout_margin="5dp"
            android:src="@drawable/back" />

        <ImageView
            android:id="@+id/image_"
            android:layout_width="250dp"
            android:layout_height="match_parent"
            android:layout_centerHorizontal="true"
            />

```

```

<TextView
    android:id="@+id/name2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_marginBottom="10dp"
    android:layout_marginLeft="15dp"
    android:text="Enchated Forest"
    android:textColor="#000000" />

```

```

<ImageView
    android:id="@+id/star"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_marginBottom="10dp"
    android:layout_marginRight="20dp"
    android:src="@drawable/empty_star"
    android:tag="0" />

```

</RelativeLayout>

总体为线性布局，图片，返回按钮，收藏按钮，以及商品名称这些使用relativelayout，将线性布局的weightsum设为3，而图片所在的relativelayout的layout_weight设为1则实现图片占整个页面的1/3。

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="0.6"
    android:background="#FFFFFF"
    android:orientation="vertical"
    android:weightSum="2">

```

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="0.99">

```

```

<TextView
    android:id="@+id/price2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="5dp"
    android:text="¥ 5.00"
    android:textColor="#D5000000" />

```

```

<TextView
    android:id="@+id/birth"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="30dp"
    android:text="作者 Johanna Basford"
    android:textColor="#8A000000" />

```

```

<View
    android:layout_width="1dip"
    android:layout_height="30dp"
    android:layout_alignParentRight="true"
    android:layout_centerVertical="true"
    android:layout_gravity="center_horizontal"
    android:layout_marginRight="60dp"
    android:background="#1E000000" />

```

```

<ImageButton
    android:id="@+id/shop2"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_alignParentRight="true"
    android:layout_centerVertical="true"
    android:layout_marginRight="20dp"
    android:background="@drawable/shoplist" />

```

</RelativeLayout>

```

<View
    android:layout_width="match_parent"
    android:layout_height="1px"
    android:layout_weight="0.02"
    android:background="#1E000000" />

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:layout_weight="0.99"
    android:divider="@drawable/shape_line"
    android:showDividers="beginning">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginLeft="15dp"
        android:text="更多产品信息"
        android:textColor="#D5000000" />
</LinearLayout>

```

```
</LinearLayout>
```

商品详情界面中部整体使用线性布局，价格购物车按钮那一行使用RelativeLayout。分割线用View实现

。

4. 点击商品列表中的item，跳转到详情界面（如何用intent传递数据）

```

mAdapter.setOnItemClickListener(new MyRecyclerViewAdapter.OnItemClickListener() {
    @Override
    public void onClick(View view, int position) {
        Map<String, Object> s = mDataas.get(position);
        Intent intent = new Intent(lab3.this, details.class);
        if (s.get("name") == "Enchated Forest") {
            //img.setImageResource(R.drawable.enchatedforest);
            Bundle bundle = new Bundle();
            bundle.putString("name2", "Enchated Forest");
            bundle.putString("price2", "¥ 5.00");
            bundle.putString("initial_1", "E");
            bundle.putString("birth", "作者 Johanna Basford");
            bundle.putInt("image", R.drawable.enchatedforest);
            intent.putExtras(bundle);

        } else if (s.get("name") == "Arla Milk") {
            Bundle bundle = new Bundle();
            bundle.putString("name2", "Arla Milk");
            bundle.putString("price2", "¥ 59.00");
            bundle.putString("initial_1", "A");
            bundle.putString("birth", "产地 德国");
            bundle.putInt("image", R.drawable.arla);
            intent.putExtras(bundle);

        } else if (s.get("name") == "Devondale Milk") {
            Bundle bundle = new Bundle();
            bundle.putString("name2", "Devondale Milk");
            bundle.putString("price2", "¥ 79.00");
            bundle.putString("initial_1", "D");
            bundle.putString("birth", "产地 澳大利亚");
            bundle.putInt("image", R.drawable.devondale);
            intent.putExtras(bundle);

        } else if (s.get("name") == "Kindle Oasis") {

```

（中间部分商

品，代码逻辑相同，不完全截图）

```

        startActivity(intent);
    }
});

```

传送数据部分（商品列表java文件）

```

Bundle bundle=this getIntent().getExtras();
tv1=(TextView)findViewById(R.id.name2);
tv1.setText(bundle.getString("name2"));
tv2=(TextView)findViewById(R.id.price2);
tv2.setText(bundle.getString("price2"));
tv3=(TextView)findViewById(R.id.birth);
tv3.setText(bundle.getString("birth"));
img=(ImageView)findViewById(R.id.image_);
img.setImageResource(bundle.getInt("image"));
ImageButton imgb=(ImageButton)findViewById(R.id.shop2);
s1= tv1.getText().toString();
s2=tv2.getText().toString();
s3=bundle.getString("initial_1");

```

接收数据部分（详情java文件）

5.长按item移除item

```

mAdapter.setOnItemClickListener(new MyRecyclerViewAdapter.OnItemClickListener() {
    @Override
    public void onLongClick(View view, int position) {
        mDatas.remove(position);
        mAdapter.notifyItemRemoved(position);
        Toast.makeText(lab3.this, "移除第"+position+"个商品", Toast.LENGTH_SHORT).show();
    }
});
mRecyclerView.setAdapter(mAdapter);

```

6.动画部分

```

mAdapter.setOnItemClickListener(new MyRecyclerViewAdapter.OnItemClickListener() {
    @Override
    public void onLongClick(View view, int position) {
        mDatas.remove(position);
        mAdapter.notifyItemRemoved(position);
        Toast.makeText(lab3.this, "移除第"+position+"个商品", Toast.LENGTH_SHORT).show();
    }
});
mRecyclerView.setAdapter(mAdapter);

```

7.购物车长按跳出提示窗口

```

        new int[] {R.id.initial_1, R.id.name_1, R.id.price});
final AlertDialog.Builder alertDialog = new AlertDialog.Builder(this);
shopList_.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {
        Map<String, Object> s = mDatas_.get(position);
        pp = position;
        String temp = s.get("name_1").toString();
        alertDialog.setTitle("移除商品").
            setMessage("从购物车移除"+temp+"?").setNegativeButton("取消", null).
            setPositiveButton("确定", (dialog, which) -> {
                mDatas_.remove(pp);
                adapter.notifyDataSetChanged();
            }).create();
        alertDialog.show();
        return true;
    }
});

```

8.购物车点击item，跳到商品详情（这部分与商品列表点击item跳到商品详情，代码实现逻辑相同，用intent传递数据）

9.商品详情页面点解back按钮，回到上一层

```

back.setOnClickListener(new ImageButton.OnClickListener() {

    @Override
    public void onClick(View v) {
        finish();
    }
});

```

10.商品详情页面，点击购物车，加到购物车，并弹出提示信息

```

imgb.setOnClickListener(new View.OnClickListener() {

    /**
     * Called when a view has been clicked.
     *
     * @param v The view that was clicked.
     */
    @Override
    public void onClick(View v) {
        Map<String, Object> map = new HashMap<>();
        map.put("initial_1", s3);
        map.put("name_1", s1);
        map.put("price", s2);
        list.add(map);
        shopList.mDatas_ = list;
        Toast.makeText(details.this, "商品已加入购物车", Toast.LENGTH_SHORT).show();
        //Toast.makeText(details.this, shopList.shop.s(), Toast.LENGTH_SHORT).show();
    }
});

```

（购物车与商品详情界面通过全局变量共享信息）

11.商品详情点击星星图标，星星图标变色


```

        final ImageView star=(ImageView)findViewById(R.id.star);
star.setTag(0);
        tag = (int)star.getTag();
star.setOnClickListener(new View.OnClickListener() {

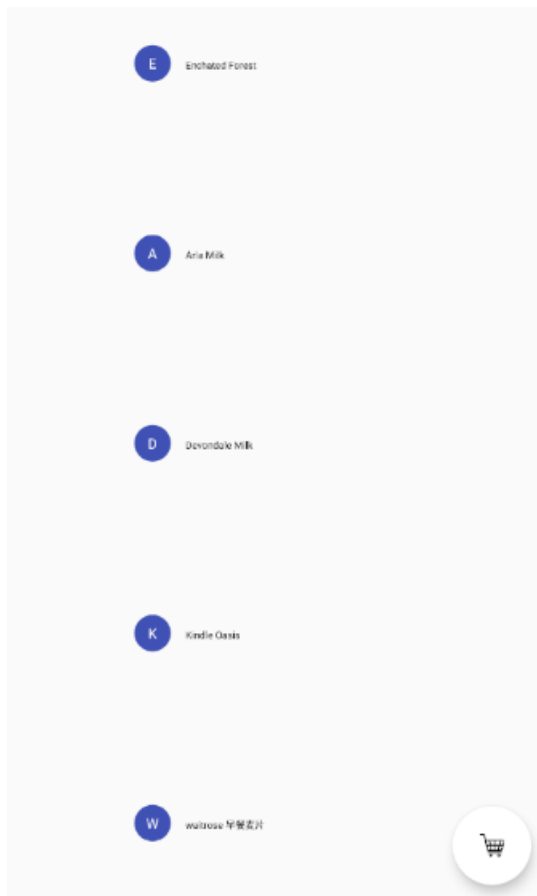
    @Override
    public void onClick(View v) {
        tag = (int)star.getTag();
        switch (tag) {
            case 0:
                star.setTag(1);
                star.setImageResource(R.drawable.full_star);
                break;
            case 1:
                star.setTag(0);
                star.setImageResource(R.drawable.empty_star);
                break;
        }
    }
});

```

(3) 实验遇到困难以及解决思路

1. 使用全局变量时出错，我使用全局变量时对全局变量不能进行除赋值之外的操作，到现在我也不知道怎么回事。
解决方法：在商品详情界面共享全局变量 list，购物车界面共享全局变量 data，每次在点击购物车按钮时都将 list 赋值给 data。
2. 星星收藏图标出错。
解决方法：首先要给星星按钮设置 tag 属性的值，才能 get 星星按钮的 tag，并复制给全局变量 tag。
3. 设置悬浮按钮时出错。
解决方法：将布局改为 RelativeLayout。
4. 添加动画依赖时出错，
解决方法：将作业阅读材料上的 2.2.7 改为适应自己的版本的 2.2.1
5. 购物车长按时也会跳到商品详情界面。
解决方法：将文档中给的返回值 false 改为 true

四、 课后实验结果



动画效果



星星图标



长按跳出删除提示

五、 实验思考及感想

这次作业代码量真的很大，而且和上次作业比难度系数差了好多，上几次作业都是课上能完成界面设计，课下半天以内就写完代码和实验报告。这次作业我写了两天时间，虽然很累，但是很有成就感，感觉一下子学到了好多知识。每个页面需要很多种布局，才能做的好看完美，是对上次布局实验的提升。这次作业的点击事件也很多，基本掌握了基础的点击事件。

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。