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 级	专业（方向）	软件工程(移动信息工程)
学号	15352033	姓名	陈黄胤
电话	18588741286	Email	chenhy98@mail2.sysu.edu.cn
开始日期	2017.10.20	完成日期	2017.10.24

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：

点击右下方的悬浮按钮可以切换到购物车：

以上两个列表点击任意一项后，可以看到详细的信息：

逻辑要求：

1.使用 RecylerView 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”

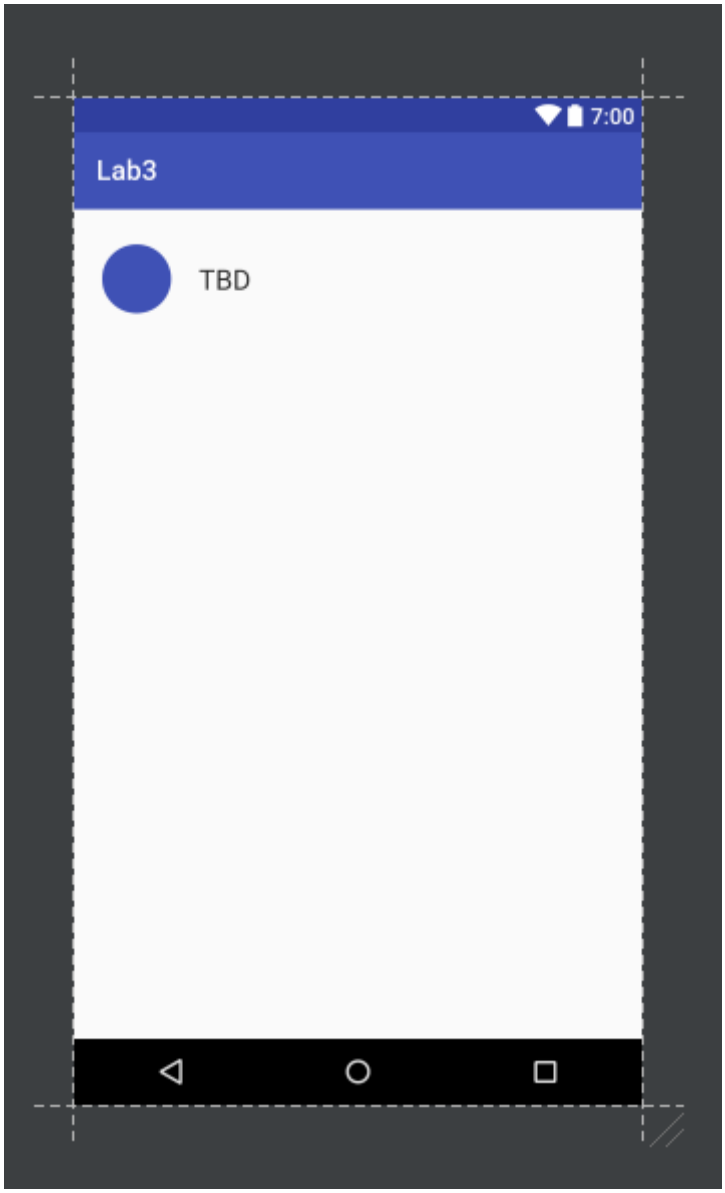
2.点击右下方的 FloatingActionButton,从商品列表切换到购物车或从购物车切换到商品列表，并且 FloatingActionButon 的图片要做相应改变。可通过设置 RecyclerView 不可见,ListView 可见来实现从商品列表切换到购物车。可通过设置 RecylerView 可见 ListView 不可见来实现从购物车切换到商品列表。

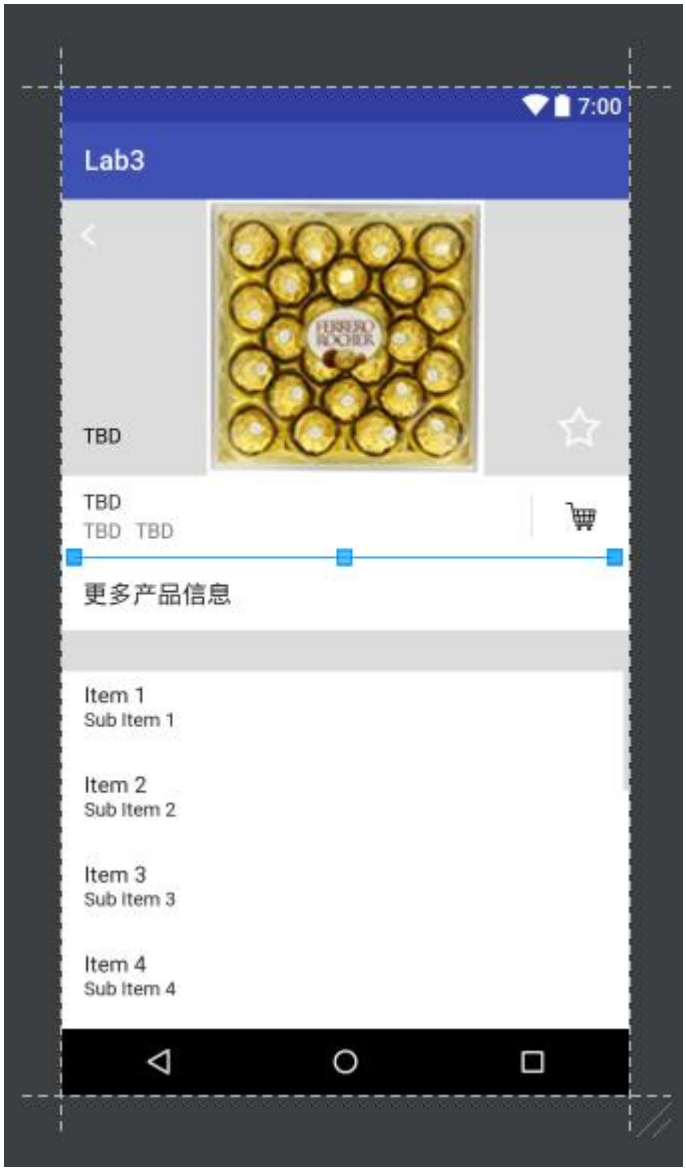
3.使用 ListView 实现购物车。点击购物车的某一个商品会跳转到商品详情界面，呈现该商品的详细信息; 长按购物中的商品会弹出对话框询问是否移除该商品，点击确定则移除该商品，点击取消则对话框消失。

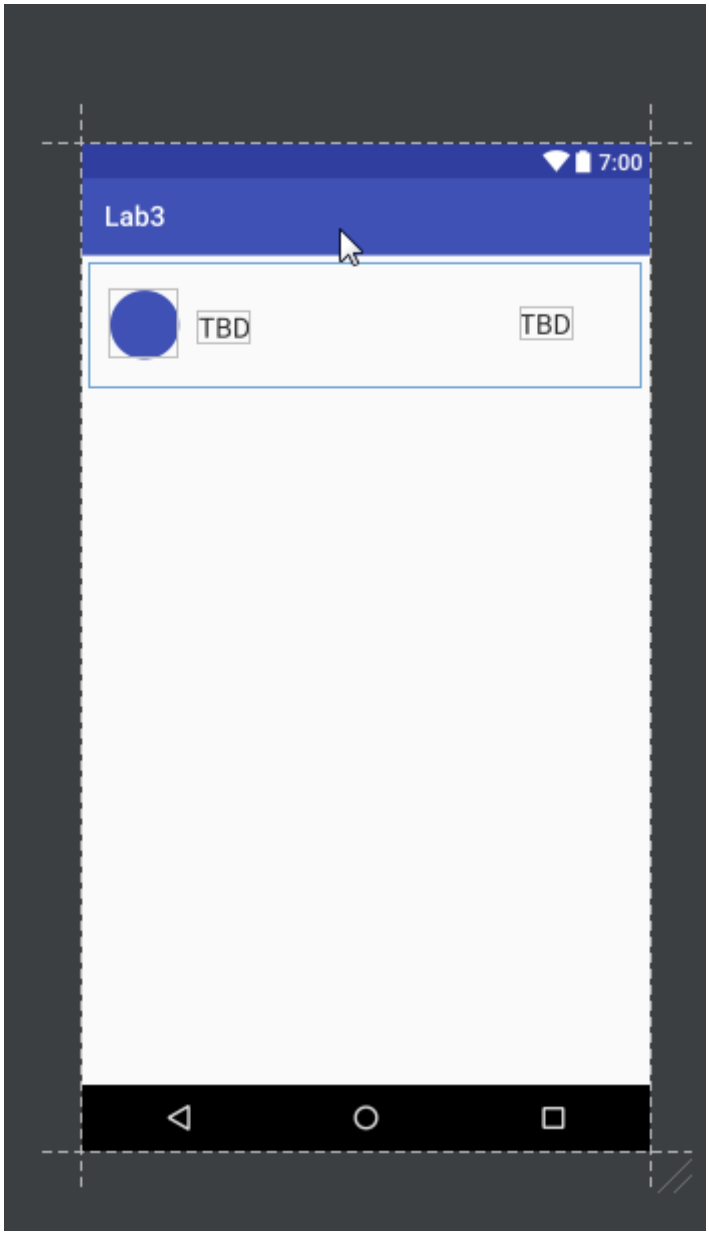
4.商品详情界面中点击返回图标会返回上一层，点击星标会切换状态，如果原先是空心星星，则会变成实心星星; 如果原先是实心星星，贝会变成空心星星。点击购物车图标会将该商品添加到购物车中并弹出 Toast 提示:"商品已添加到购物车"。不要求判断购物车是否已有该商品,即如果已有一件该商品,添加之后则显示两个即可。未退出商品详细界面时点击多次购物车图标可以只添加一件商品也可以添加多件到购物车中。

三、 课堂实验结果

（1） 实验截图


7:00
Lab3
TBD


7:00
Lab3
TBD
TBD
TBD TBD
更多产品信息
Item 1
Sub Item 1
Item 2
Sub Item 2
Item 3
Sub Item 3
Item 4
Sub Item 4



（2） 实验步骤以及关键代码

Ⅰ.设计布局

```
<ListView
    android:id="@+id/shoppinglist"
    android:visibility="gone"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    />

<android.support.v7.widget.RecyclerView
    android:id="@+id/recycler_view"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginStart="8dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    />
<android.support.design.widget.FloatingActionButton
    android:id="@+id/shoplist"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="40dp"
    android:src="@drawable/shoplist"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="50dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="1.0"

    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0"
    app:rippleColor="@color/white"
    app:backgroundTint="@color/white"
     />
</android.support.constraint.ConstraintLayout>
```

如上图所示，设计第一个页面，包括一个 RecyclerView 用于展示商品列表，一个 ListView 用于展示购物车列表，一个 FloatingActionButton 控制上述两个模块的 Visibility 以及购物车切换事件。

```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <RelativeLayout
        android:background="@color/gray"
        android:layout_width="fill_parent"
        android:layout_height="0dp"
        android:layout_weight="1.0">
        <ImageView
            android:id="@+id/top_back"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:src="@drawable/ferrero"
            />
        <ImageView
            android:id="@+id/back"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="10dp"
            android:layout_marginTop="10dp"
            android:src="@drawable/back"
            />
        <TextView
            android:textSize="15.0sp"
            android:textColor="#ff000000"
            android:id="@+id/name"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="15dp"
            android:layout_marginBottom="20dp"
            android:text="TBD"
            android:layout_alignParentBottom="true"
            />
        <ImageView
            android:id="@+id/star"
            android:background="@drawable/empty_star"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="20dp"
```

```
        />
    <ImageView
        android:id="@+id/star"
        android:background="@drawable/empty_star"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="20dp"
        android:layout_alignParentEnd="true"
        />
</RelativeLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="0dp"
    android:layout_weight="2.0"
    android:background="@color/gray"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@color/white"
        android:orientation="vertical">

        <RelativeLayout
            android:layout_width="fill_parent"
            android:layout_height="0dp"
            android:layout_weight="1.0"
            >

            <TextView
                android:id="@+id/price"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="15dp"
                android:layout_marginTop="10dp"
                android:text="TBD"
                android:textColor="#d5000000"
                android:textSize="15.0sp" />

            <LinearLayout
```

```
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/price"
        android:layout_marginBottom="10dp"
        android:layout_marginStart="15dp"
        android:layout_marginTop="3dp">

        <TextView
            android:id="@+id/type"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="10dp"
            android:text="TBD"
            android:textColor="#8a000000"
            android:textSize="15.0sp" />

        <TextView
            android:id="@+id/info"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="TBD"
            android:textColor="#8a000000"
            android:textSize="15sp" />
    </LinearLayout>

    <ImageView
        android:layout_width="1dp"
        android:layout_height="0dp"
        android:layout_alignBottom="@+id/buy"
        android:layout_alignTop="@+id/buy"
        android:layout_centerVertical="true"
        android:layout_toStartOf="@+id/buy"
        android:background="@color/gray" />

    <ImageView
        android:id="@+id/buy"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
```

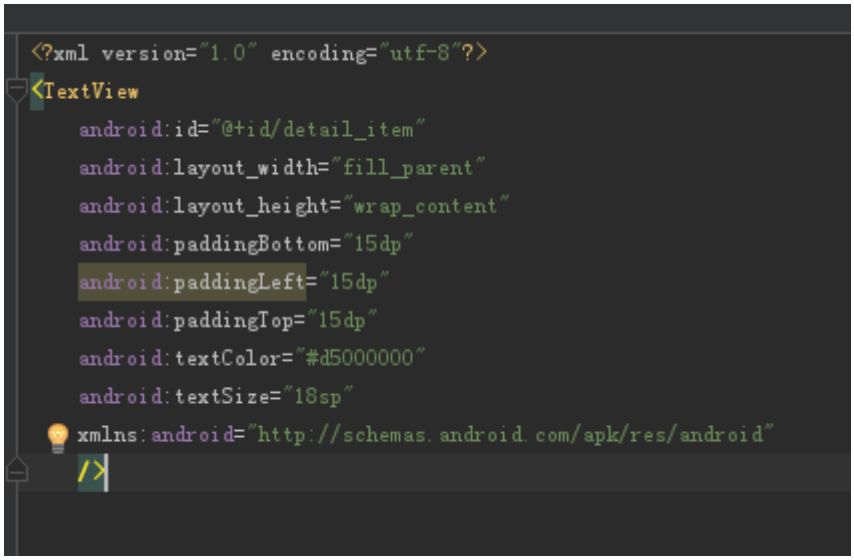
```
            android:background="@color/gray" />

        <ImageView
            android:id="@+id/buy"
            android:layout_width="30dp"
            android:layout_height="30dp"
            android:layout_alignParentEnd="true"
            android:layout_centerVertical="true"
            android:layout_marginLeft="20dp"
            android:layout_marginRight="20dp"
            android:src="@drawable/shoplist" />
    </RelativeLayout>

    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="1dp"
        android:layout_marginEnd="10dp"
        android:layout_marginStart="10dp"
        android:background="@color/gray" />

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="15dp"
        android:paddingLeft="15dp"
        android:paddingTop="15dp"
        android:text="更多产品信息"
        android:textColor="#d5000000"
        android:textSize="18sp" />
</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    >

</LinearLayout>

<ListView
    android:id="@+id/operation_list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
```

如上图所示设计了商品的详细界面，采用的是 LinearLayout。并定义了一系列的子布局，包括商品图片，商品名称，收藏星星位于第一个 RelativeLayout 作为 LinearLayout 的第一个子布局，使用 layout_weight 来定义子布局的占位大小。然后再定义一个 LinearLayout 作为第二个子布局，占位 2/3。子布局 LinearLayout 中再分子布局，使用 image 来画分隔线，并最后采用 ListView 来实现商品的信息列表。

除了上面所述的两个页面布局设计，还为每个 ListView 以及 RecyclerView 制定了布局，分别为商品列表 RecyclerView 的每一项的布局、购物车列表 ListView 的每一项的布局，商品信息 ListView 中的每一项布局，代码分别如下图所示：

```
<android.support.constraint.ConstraintLayout
    android:color="#1e000000"
    android:drawable="@color/white"
    android:paddingTop="5dp"
    android:paddingBottom="5dp"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <TextView
        android:textSize="25sp"
        android:textColor="#ffffff"
        android:gravity="center"
        android:id="@+id/first1"
        android:background="@drawable/icon1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="15dp"
        android:layout_marginTop="8dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        />
    <TextView
        android:textSize="20sp"
        android:textColor="#d5000000"
        android:layout_gravity="center"
        android:id="@+id/list_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="8dp"
        android:layout_marginStart="20dp"
        android:text="TBD"
        app:layout_constraintBottom_toBottomOf="@+id/first1"
        app:layout_constraintLeft_toRightOf="@+id/first1"
        app:layout_constraintTop_toTopOf="@+id/first1"
        />
```

```xml
<?xml version="1" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    android:color="#1e000000"
    android:drawable="@color/white"
    android:paddingTop="5dp"
    android:paddingBottom="5dp"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <TextView android:textSize="25sp"
        android:textColor="#ffffffff"
        android:gravity="center"
        android:id="@+id/first"
        android:background="@drawable/icon1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="15dp"
        android:layout_marginTop="8dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginLeft="16dp"
        app:layout_constraintVertical_bias="0.48" />
    <TextView
        android:textSize="20sp"
        android:textColor="#d5000000"
        android:layout_gravity="center"
        android:id="@+id/shopcar_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="13dp"
        android:layout_marginBottom="8dp"
        android:layout_marginStart="14dp"
        android:text="TBD"
        app:layout_constraintBottom_toBottomOf="@+id/first"
        app:layout_constraintLeft_toRightOf="@+id/first"
        app:layout_constraintTop_toTopOf="@+id/first"
        />
    <TextView
        android:textSize="20sp"
        android:textColor="#d5000000"
        android:layout_gravity="center"
        android:id="@id/price"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="13dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="52dp"
        android:text="TBD"
        app:layout_constraintBottom_toBottomOf="@id/first"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="@id/first"
        app:layout_constraintVertical_bias="0"

        />
</android.support.constraint.ConstraintLayout>
```


<?xml version="1.0" encoding="utf-8"?>
<TextView
android:id="@+id/detail_item"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingBottom="15dp"
android:paddingLeft="15dp"
android:paddingTop="15dp"
android:textColor="#d5000000"
android:textSize="18sp"
xmlns:android="http://schemas.android.com/apk/res/android"
/>

Ⅱ.设计 java 事件

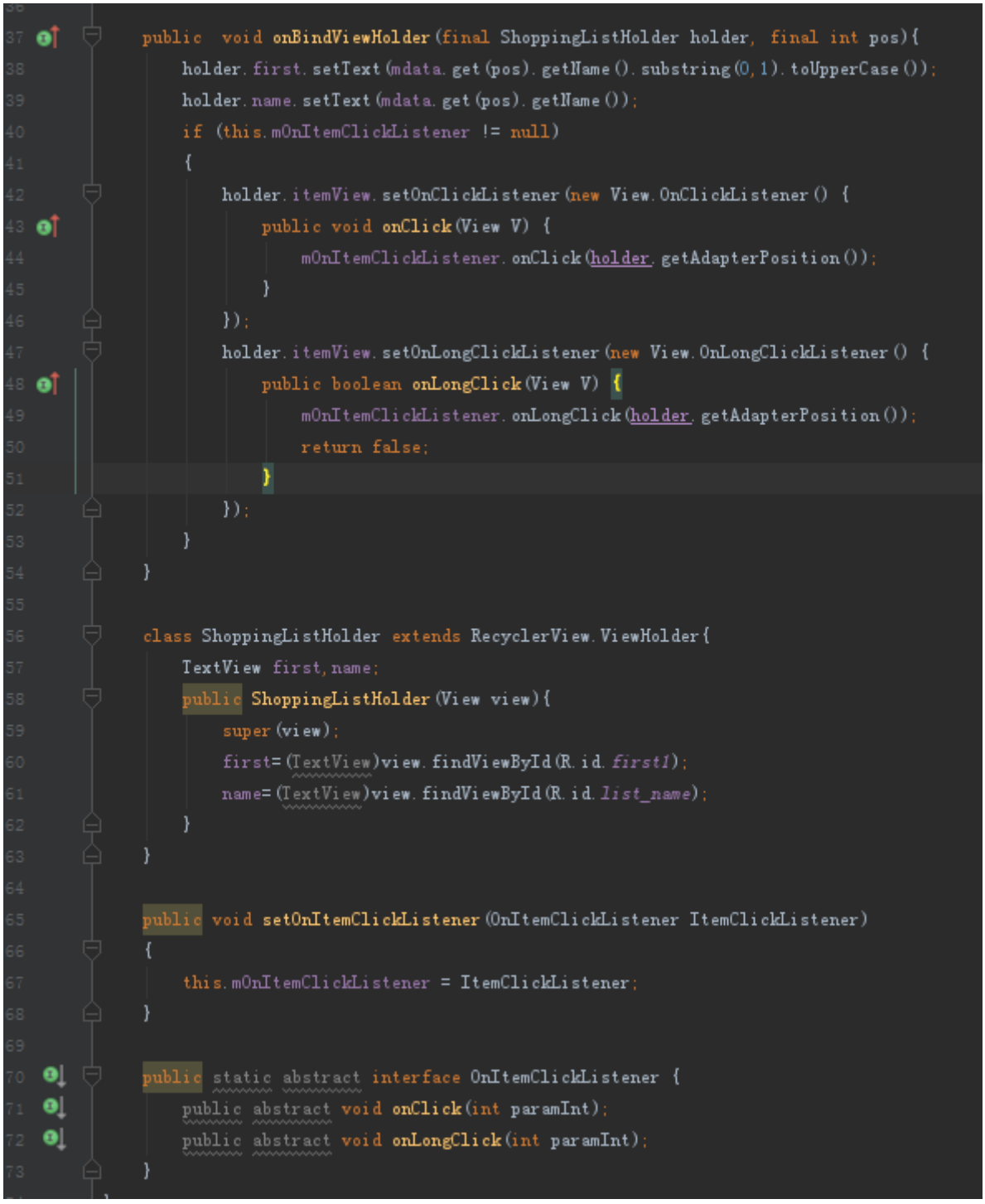
```java
public class ShoppingCarListAdapter extends BaseAdapter
{
    private Context context;
    private List<Goods_data> dataList;

    public ShoppingCarListAdapter(List<Goods_data> List,Context Context)
    {
        this.context = Context;
        this.dataList = List;
    }

    private boolean isNull() { return this.dataList == null; }

    @Override
    public int getCount()
    {
        if (isNull())
            return 0;
        return this.dataList.size();
    }

    public List<Goods_data> getDataList() { return this.dataList; }

    @Override
    public Object getItem(int pos)
    {
        if (isNull())
            return null;
        return this.dataList.get(pos);
    }

    @Override
    public long getItemId(int paramInt) { return paramInt; }
```

```
@Override
public long getItemId(int paramInt) { return paramInt; }

@Override
public View getView(int pos, View view, ViewGroup paramViewGroup)
{
    View localView;
    ViewHolder localViewHolder;
    if (view == null)
    {
        localView = LayoutInflater.from(this.context).inflate(R.layout.shopping_car, null);
        localViewHolder = new ViewHolder();
        localViewHolder.image = ((TextView)localView.findViewById(R.id.first));
        localViewHolder.name = ((TextView)localView.findViewById(R.id.shopcar_name));
        localViewHolder.price = ((TextView)localView.findViewById(R.id.price));
        localView.setTag(localViewHolder);
    }
    else
    {
        localView = view;
        localViewHolder = (ViewHolder)localView.getTag();
    }
    localViewHolder.image.setText(dataList.get(pos).getName().substring(0,1).toUpperCase());
    if(dataList.get(pos).getName() == "购物车") {
        localViewHolder.image.setText("*");
    }
    localViewHolder.name.setText(dataList.get(pos).getName());
    localViewHolder.price.setText(dataList.get(pos).getPrice());
    return localView;
}


private class ViewHolder
{
    public TextView image;
    public TextView name;
    public TextView price;
    private ViewHolder(){
    }
}
```

如上图所示，设计购物车 ListView 的 Adapter:

两个 private 变量 context 和 dataList，分别用来存储上下文以及购物车中的对象 List 然后重载 BaseAdapter 中的函数即可。这里还设计了 ViewHolder，三个 TextView 分别代表购物车中每一项物品的三个 TextView，并在 getView 上进行了 id 关联。

```
public class ShoppingListAdapter extends RecyclerView.Adapter<ShoppingListAdapter.ShoppingListHolder>{
    private List<Goods_data> mdata;
    private OnItemClickListener mOnItemClickListener = null;
    private Context context;

    public ShoppingListAdapter(List<Goods_data> mdata,Context context){
        this.context = context;
        this.mdata = mdata;
    }

    @Override
    public int getItemCount() { return this.mdata.size(); }

    public ShoppingListHolder onCreateViewHolder(ViewGroup viewGroup, int View_type){
        View view= LayoutInflater.from(context).inflate(R.layout.editable_item,viewGroup,false);
        return new ShoppingListHolder(view);
    }

    public  void onBindViewHolder(final ShoppingListHolder holder, final int pos){
        holder.first.setText(mdata.get(pos).getName().substring(0,1).toUpperCase());
        holder.name.setText(mdata.get(pos).getName());
        if (this.mOnItemClickListener != null)
        {
            holder.itemView.setOnClickListener((V) -> {
                    mOnItemClickListener.onClick(holder,getAdapterPosition());
            });
            holder.itemView.setOnLongClickListener((V) -> {
                    mOnItemClickListener.onLongClick(holder,getAdapterPosition());
                    return false;
            });
        }
    }

    class ShoppingListHolder extends RecyclerView.ViewHolder{
        TextView first,name;
        public ShoppingListHolder(View view){
            super(view);
            first=(TextView)view.findViewById(R.id.first1);
```



```
public  void onBindViewHolder(final ShoppingListHolder holder, final int pos){
    holder.first.setText(mdata.get(pos).getName().substring(0,1).toUpperCase());
    holder.name.setText(mdata.get(pos).getName());
    if (this.mOnItemClickListener != null)
    {
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            public void onClick(View V) {
                mOnItemClickListener.onClick(holder.getAdapterPosition());
            }
        });
        holder.itemView.setOnLongClickListener(new View.OnLongClickListener() {
            public boolean onLongClick(View V) {
                mOnItemClickListener.onLongClick(holder.getAdapterPosition());
                return false;
            }
        });
    }
}

class ShoppingListHolder extends RecyclerView.ViewHolder{
    TextView first,name;
    public ShoppingListHolder(View view){
        super(view);
        first=(TextView)view.findViewById(R.id.first1);
        name=(TextView)view.findViewById(R.id.list_name);
    }
}

public void setOnItemClickListener(OnItemClickListener ItemClickListener)
{
    this.mOnItemClickListener = ItemClickListener;
}

public static abstract interface OnItemClickListener {
    public abstract void onClick(int paramInt);
    public abstract void onLongClick(int paramInt);
}
```

如上图所示，设计商品列表 RecyclerView 的 adapter，只需要按要求重载 RecyclerView.adapter 的 getItemCount、onBindViewHolder、onCreateViewHolder 这几个函数即可，使用的 ViewHolder 直接继承 RecylerView 类的成员 ViewHolder，在里面定义两个 TextView，在构造函数中分别绑定到商品列表的首字母以及商品名字的布局 id 上。BindViewHolder 则将内容赋予给 holder，内容包括两个 text 以及点击、长点击的属性。

```
public class Goods_list extends AppCompatActivity {

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        findAllViews();
        load_data();
        set_Listener();
    }

    private ListView shoppingList;
    private RecyclerView mRecyclerView;
    private FloatingActionButton mFAButton;
    private AlertDialog.Builder mbuilder;
    private List<Goods_data> data = new ArrayList<>();
    private List<Goods_data> shopping_list = new ArrayList<>();
    final ShoppingListAdapter mAdapter = new ShoppingListAdapter(data, Goods_list
    final ShoppingCarListAdapter CarAdapter = new ShoppingCarListAdapter(shopping

    final String[] product_name = new String[] {"Enchated Forest","Arla Milk","De
    final String[] product_price = new String[] {"￥5.00","￥59.00","￥79.00","￥
    final String[] product_info_type = new String[] {"作者","产地","产地","版本",
    final String[] product_info = new String[] {"   Johanna Basford","   德国","
    final int[] imageID = new int[]{R.drawable.enchatedforest,R.drawable.arla,R.d
```

设计第一个页面的 activity，首先先定义了一系列要用到的变量，然后打表将要显示的内容先用变量保存起来。

```
private void findAllViews() {
    setContentView(R.layout.home_page);
    this.shoppingList = ((ListView)findViewById(R.id.shoppinglist));
    this.mRecyclerView = ((RecyclerView)findViewById(R.id.recycler_view));
    this.mbuilder = new AlertDialog.Builder(this);
    this.mFAButton = ((FloatingActionButton)findViewById(R.id.shoplist));
}
```

将需要用到的 id 绑定到变量上。

```
    {
        for(int i=0;i<9;i++) {
            this.data.add(new Goods_data(product_name[i],product_pr
        }
        Goods_data temp = new Goods_data("购物车", "价格   ", null,
        shopping_list.add(temp);
        shoppingList.setAdapter(CarAdapter);
        mRecyclerView.setLayoutManager(new LinearLayoutManager(this
        mAdapter.setOnItemClickListener(new ShoppingListAdapter.OnI
            @Override
```

添加完内容后绑定对应的 adpter

```
@Override
public void onClick(int position) {
        Intent intent = new Intent(Goods_list.this, Goods_detail.class);
        Goods_data deliver = new Goods_data(data.get(position).getName(), data.get(position).getPrice(), data.
        Bundle localBundle = new Bundle();
        localBundle.putSerializable("goods", deliver);
        intent.putExtras(localBundle);
        Goods_list.this.startActivityForResult(intent, 1);
}
@Override
public void onLongClick(int position) {
    Toast.makeText(Goods_list.this,"移除第"+String.valueOf(position+1)+"个商品",Toast.LENGTH_SHORT).show()
    data.remove(position);
    mAdapter.notifyDataSetChanged();
}
```

实现商品列表的功能，点击出现商品详情，长按删除商品并发出 Toast，两个活动的联络使用 Intent

```
        });
        //mRecyclerView.setAdapter(mAdapter);
        ScaleInAnimationAdapter animationAdapter = new ScaleInAnimationAdapter(mAdapter);
        animationAdapter.setDuration(1000);
        mRecyclerView.setAdapter(animationAdapter);
        mRecyclerView.setItemAnimator(new OvershootInLeftAnimator());
    }
```

设置动画 adapter

```
private void set_Listener(){
    this.mFAButton.setOnClickListener((paramAnonymousView) -> {
            if (Goods_list.this.mRecyclerView.getVisibility() == View.INVISIBLE)
            {
                Goods_list.this.mRecyclerView.setVisibility(View.VISIBLE);
                Goods_list.this.shoppingList.setVisibility(View.INVISIBLE);
                Goods_list.this.mFAButton.setImageResource(R.drawable.shoplist);
                return;
            }
            Goods_list.this.mRecyclerView.setVisibility(View.INVISIBLE);
            Goods_list.this.shoppingList.setVisibility(View.VISIBLE);
            Goods_list.this.mFAButton.setImageResource(R.drawable.mainpage);
    });
    this.mbuilder.setNegativeButton("取消", (AnonymousDialogInterface, paramAnonymousInt) -> {
            Toast.makeText(getApplicationContext(), "你选择了[取消]", Toast.LENGTH_SHORT).show();
    });

    this.shoppingList.setOnItemClickListener((AdapterView, view, i, l) -> {
    this.shoppingList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> AdapterView, View view, int i, long l) {
            Intent localIntent = new Intent(Goods_list.this, Goods_detail.class);
            Bundle localBundle = new Bundle();
            localBundle.putSerializable("goods", (Serializable)Goods_list.this.shopping_list.get(i));
            localIntent.putExtras(localBundle);
            Goods_list.this.startActivityForResult(localIntent, 1);
```

设置页面的 visibility 事件

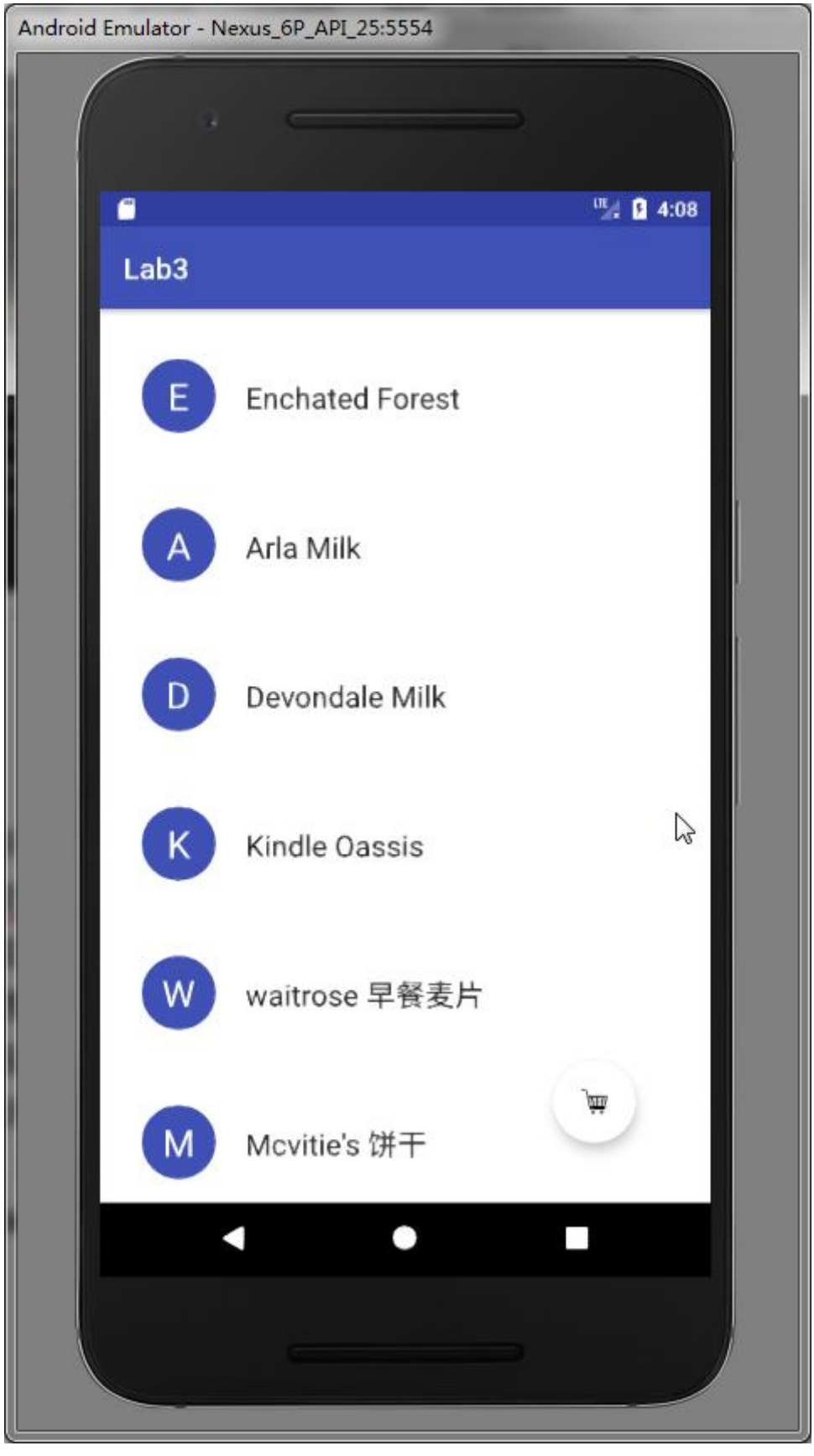
```
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
        if(requestCode == 1 && resultCode ==1){
            Goods_data localGoods = (Goods_data)intent.getExtras().get("goods");
            this.shopping_list.add(localGoods);
            CarAdapter.notifyDataSetChanged();
        }
    }
```

商品详情处的添加购物车事件，设置信息接收函数 onActivityResult

```
private void initialData()
{
    Resources localResources = getResources();
    this.goods = ((Goods_data) getIntent().getExtras().get("goods"));
    if (this.goods != null)
    {
        this.topBack.setImageResource(this.goods.getImageId());
        this.name.setText(this.goods.getName());
        this.price.setText(this.goods.getPrice());
        this.type.setText(this.goods.getType());
        this.info.setText(this.goods.getInfomation());
    }
    String[] arrayOfString = new String[4];
    arrayOfString[0] = localResources.getString(R.string.item1);
    arrayOfString[1] = localResources.getString(R.string.item2);
    arrayOfString[2] = localResources.getString(R.string.item3);
    arrayOfString[3] = localResources.getString(R.string.item4);
    ArrayAdapter localArrayAdapter = new ArrayAdapter<>(this, R.layout.detail_list_i
    this.operationList.setAdapter(localArrayAdapter);
    this.star.setTag("0");
}
```

商品详细信息事件的内容输入。

四、 课后实验结果


Android Emulator - Nexus_6P_API_25:5554
4:08
Lab3
E
Enchated Forest
A
Arla Milk
D
Devondale Milk
K
Kindle Oassis
W
waitrose 早餐麦片
M
Mcvitie's 饼干


Android Emulator - Nexus_6P_API_25:5554
4:08
Lab3
Ferrero Rocher
¥ 132.59
重量 300g
更多产品信息
一键下单
分享商品
不感兴趣
查看更多商品促销信息


Android Emulator - Nexus_6P_API_25:5554
4:09
Lab3
Ferrero Rocher
¥ 132.59
重量 300g
更多产品信息
一键下单
分享商品
不感兴趣
商品已加到购物车
查看更多商品促销信息


Android Emulator - Nexus_6P_API_25:5554
4:09
Lab3
*
购物车
价格
F
Ferrero Rocher
¥ 132.59


Android Emulator - Nexus_6P_API_25:5554
4:09
Lab3
*
购物车
价格
F
Ferrero Rocher
¥ 132.59
从购物车移除Ferrero Rocher?
取消
确定

五 、**实验思考及感想**

这一次的实验内容很多，需要很熟悉事件间的信息传递方法，比如这一次的传递，如果一个变量一个变量地传递会很麻烦，putExtra 又有各种

各样的限制，就想着能不能直接传递对象，上网查资料得知，可以使用 Serializable 来使得对象可被传递，这样一来每一个商品的信息都能被很完整地被传递，因为每次传递都是以对象为单位。除此之外，还有许多需要注意的细节，例如每一个 activity 都需要在 manifest 里面注册，不然会导致安卓程序崩溃。细节很重要，一定要熟悉语法，不然几个布局几个页面堆起来会导致 debug 非常困难，这也算是这一次实验的一点经验所得。