

# 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：

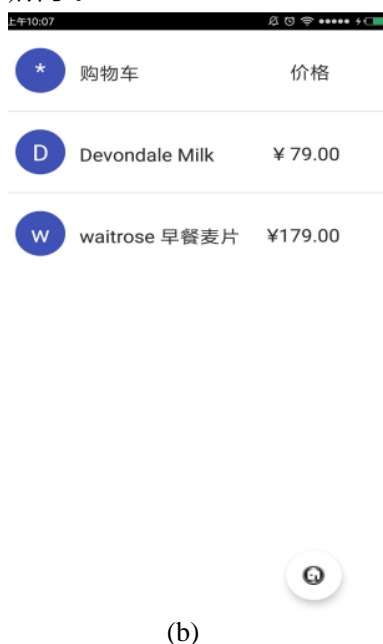
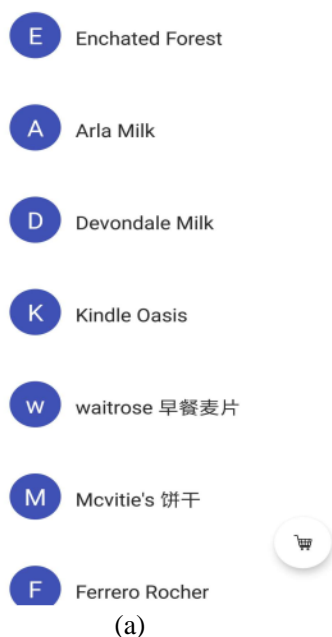
年级	15352272	专业 ( 方向 )	软件工程 ( 移动信息工程 )
学号	15352272	姓名	彭国栋
电话	15626066058	Email	578291308@qq.com
开始日期		完成日期	

### 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，如图(a)所示，而点击右下方的悬浮按钮可以切换到购物车，如图(b)所示。

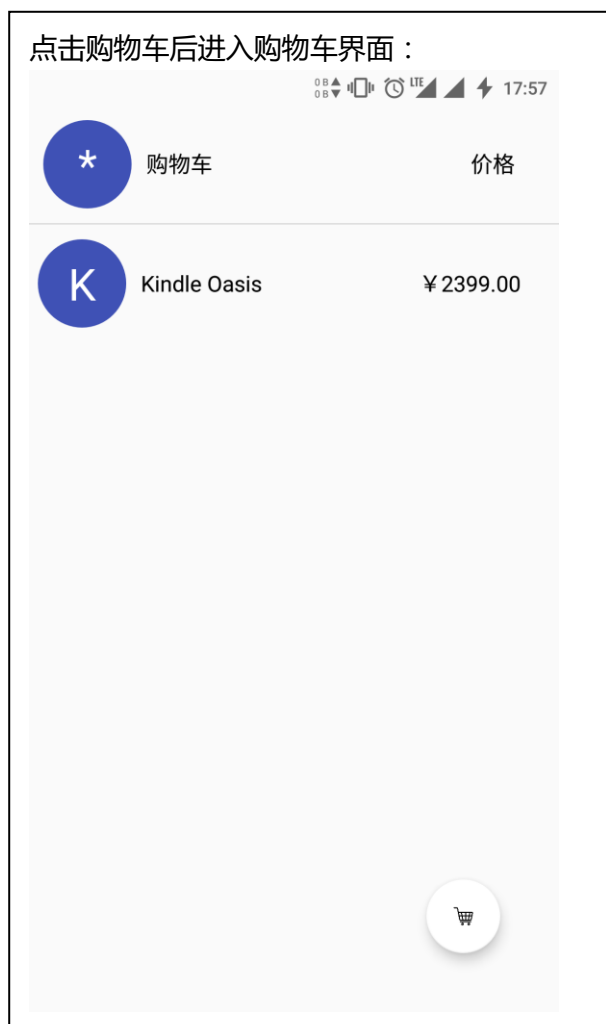
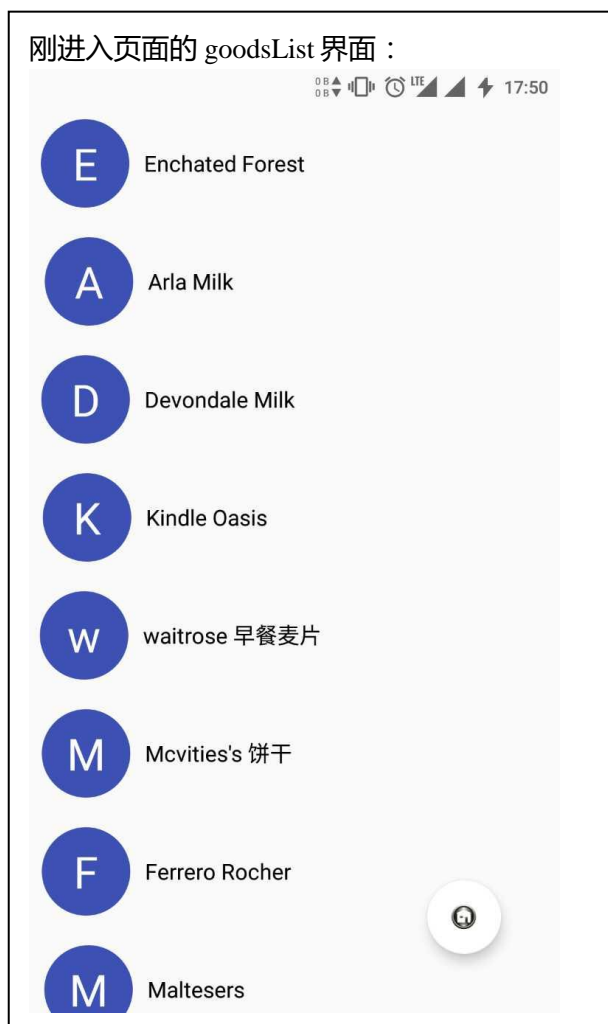


上面两个列表点击任意一项后，可以看到详细的信息：



### 三、 课堂实验结果

#### (1) 实验截图



进入详情之后未收藏界面：



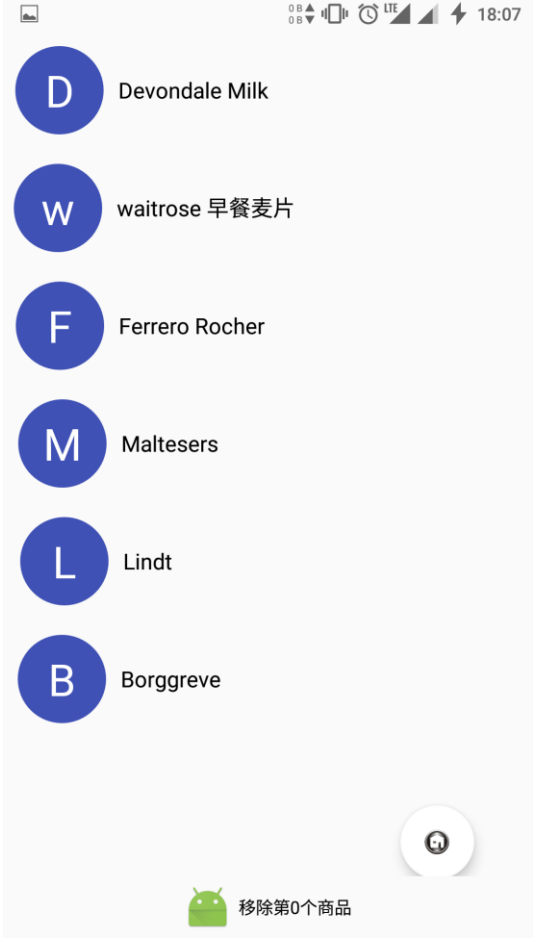
进入详情之后收藏界面：



购物车界面长按删除：

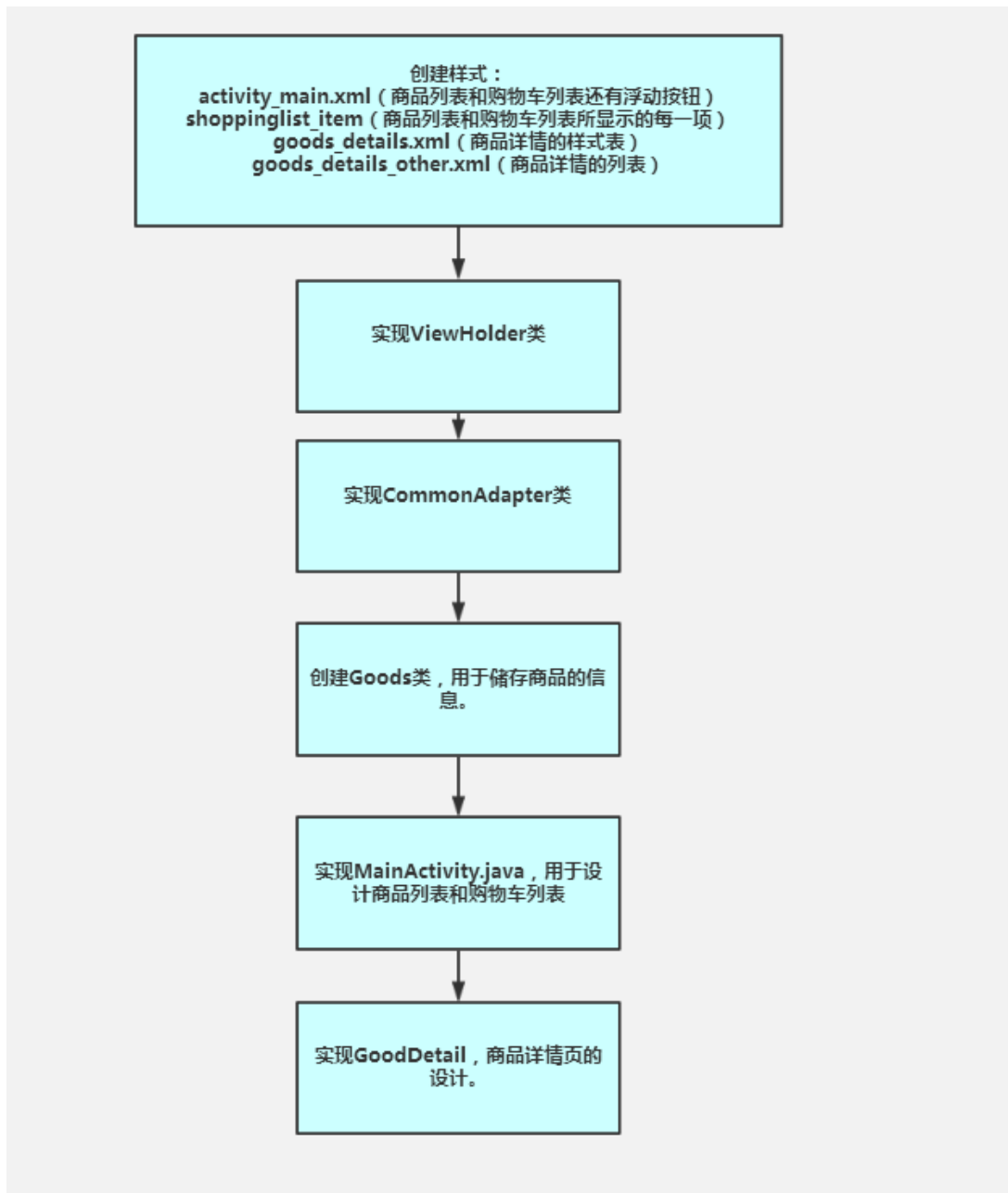


进入详情之后收藏界面：



## (2) 实验步骤以及关键代码

### 1、实验步骤：



### 2、实验代码：

- 样式部分：

activity\_main.xml:商品列表和购物车列表。

ListView, 购物车列表详情。

RecyclerView, 商品列表详情。

FloatingActionButton, 浮动按钮。

```
<ListView
    android:id="@+id/shoppinglist"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    >
</ListView>

<android.support.design.widget.FloatingActionButton
    android:id="@+id/shoppingcar"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:src="@mipmap/mainpage"
    android:layout_marginBottom="40dp"
    android:layout_marginRight="40dp"

    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:backgroundTint="@color/colorWhite"
    android:layout_marginEnd="40dp" />
</android.support.constraint.ConstraintLayout>

<android.support.v7.widget.RecyclerView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/mrecyclerview"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    >
</android.support.v7.widget.RecyclerView>
```

shoppinglis\_item.xml:商品列表和购物车列表项。

3 个 TextView, 分别为: 背景为蓝色圆形的显示商品名字首字母。商品名字的显示位置。商品价格的显示位置。

```
<TextView
    android:id="@+id/goodsFirst_letter"
    android:layout_width="60dp"
    android:layout_height="60dp"
    android:textSize="30dp"
    android:background="@drawable/background_shape"
    android:gravity="center"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp"
    android:textColor="@color/colorWhite"
    />

<TextView
    android:id="@+id/price"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/colorBlack"
    android:textSize="15dp"
    android:paddingRight="20dp"
    android:gravity="right"
    android:layout_gravity="center"
    android:layout_weight="0.5" />

<TextView
    android:id="@+id/goodsName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:paddingLeft="10dp"
    android:textSize="15dp"
    android:textColor="@color/colorBlack"
    android:text="gouwu"
    android:layout_weight="0.4"
    />
```

goods\_details.xml: 商品详情列表的设计

RelativeLayout: 包括 3 个 ImageView 和 1 个 TextView, 3 个 ImageView 用于显示返回、星星和商品图片。

View: 分割线若干, 这里不细看。

TextView: 显示产地重量等类型信息, 和显示具体属性的信息, 还有显示更多商品信息的

ListView: 显示"一键下单","分享商品","不感兴趣","查看更多商品信息"的列表。

```
<RelativeLayout
    android:id="@+id/relativeLayoutForImage"
    android:layout_width="0dp"
    android:layout_height="200dp"

    android:background="@drawable/background_relativelayout"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
>
    <ImageView
        android:id="@+id/goodsImage"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:src="@mipmap/ar1a"

        android:layout_centerHorizontal="true"
    />

    <TextView
        android:id="@+id/detail_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="test"
        android:layout_alignBottom="@id/goodsImage"
        android:layout_alignParentLeft="true"
        android:paddingLeft="10dp"
        android:paddingBottom="10dp"
    />

    <TextView
        android:id="@+id/detail_price"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="15dp"
        android:textColor="@color/colorBlack"
        app:layout_constraintTop_toBottomOf="@+id/relativeLayoutForImage"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginTop="10dp"
        android:text="¥89"
        android:layout_marginLeft="10dp"
        android:layout_marginStart="10dp" />

    <TextView
        android:id="@+id/detail_type"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="15dp"
        android:text="产地"
        android:textColor="#8A000000"
        app:layout_constraintTop_toBottomOf="@+id/detail_price"
        android:layout_marginTop="10dp"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginLeft="10dp"
        android:layout_marginStart="10dp" />

    <TextView
        android:id="@+id/detail_moreDetails"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20dp"
        android:text="更多产品信息"
        android:textColor="@color/colorBlack"
        app:layout_constraintTop_toBottomOf="@+id/divide_line2"
        android:layout_marginTop="10dp"
        app:layout_constraintLeft_toLeftOf="parent"
        android:layout_marginLeft="10dp"
        android:layout_marginStart="10dp" />

    <ImageView
        android:id="@+id/star"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignBottom="@id/goodsImage"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="10dp"
        android:src="@mipmap/empty_star"/>

    <ImageView
        android:id="@+id/backbutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignTop="@id/goodsImage"
        android:src="@mipmap/back"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="10dp"
    />

    <TextView
        android:id="@+id/detail_info"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="111"
        app:layout_constraintLeft_toRightOf="@+id/detail_type"
        app:layout_constraintTop_toBottomOf="@+id/detail_price"
        android:textSize="15dp"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginStart="10dp" />

    <ImageView
        android:id="@+id/detail_shoppingcar"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:src="@mipmap/shoplist"
        app:layout_constraintTop_toBottomOf="@+id/relativeLayoutForImage"
        app:layout_constraintRight_toRightOf="parent"
        android:layout_marginTop="15dp"
        android:layout_marginRight="20dp"
        android:layout_marginEnd="20dp" />

    <ListView
        android:id="@+id/othersList"
        android:layout_height="wrap_content"
        android:layout_width="0dp"
        app:layout_constraintTop_toBottomOf="@+id/divide_line3"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
    />
```

goods\_details\_other.xml: 显示"一键下单","分享商品","不感兴趣","查看更多商品信息"的列表项。使用一个 TextView 实现:

```
<TextView
    android:id="@+id/others"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="gengduo"
    android:textSize="20dp"
    android:paddingLeft="10dp"
    android:layout_gravity="center"
    android:textColor="@color/colorBlack"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
/>
```

- java 程序部分：

①、ViewHolder 类：ViewHolder 通常出现在适配器里，为的是 listview 滚动的时候快速设置值，而不必每次都重新创建很多对象，从而提升性能。在这里我们通过继承 RecyclerView.ViewHolder 类的方式，实现 ViewHolder 类。

```
public class ViewHolder extends RecyclerView.ViewHolder {
    private SparseArray<View> mViews;
    private View mListview;
    public ViewHolder(Context context, View itemView, ViewGroup parent)
    { // 默认构造函数

        super(itemView);
        mListview=itemView;
        mViews=new SparseArray<View>();
    }

    public static ViewHolder get(Context context,ViewGroup parent,int layoutid)
    {
        View itemView= LayoutInflater.from(context).inflate(layoutid,parent
        ,false); //inflate 可以将一个xml 中定义的布局控件找出来
        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)
        {
            view=mListview.findViewById(viewId);
            mViews.put(viewId,view);
        }
        return (T) view;
    }
}
```

②、CommoAdapter 类：用于作为

```
public abstract class CommonAdapter<T> extends RecyclerView.Adapter<ViewHolder>{
    protected Context mContext;
    protected int mLayoutId;
    protected List<T> mDatas;
    private OnItemClickListener mOnItemClickListener=null;
    public CommonAdapter(Context context,int layoutId,List<T> datas)
    { //构造函数
        mContext=context;
        mLayoutId=layoutId;
        mDatas=datas;
    }
    @Override
    public ViewHolder onCreateViewHolder(final ViewGroup parent, int viewType)
    { //创建ViewHolder
        ViewHolder viewHolder=ViewHolder.get(mContext,parent,mLayoutId);
        return viewHolder;
    }

    @Override
    public void onBindViewHolder(final ViewHolder holder,int position)
    { //这个方法主要用于适配渲染数据到View中。方法提供给你一个viewHolder，而不是原来的convertView。
        convert(holder,mDatas.get(position));
        if(mOnItemClickListener!=null)
        {
            holder.itemView.setOnClickListener((v) -> {
                mOnItemClickListener.onClick(holder.getAdapterPosition());
            });
            holder.itemView.setOnLongClickListener((v) -> {
                mOnItemClickListener.onLongClick(holder.getAdapterPosition());
                return false;
            });
        }
    }
    public abstract void convert(ViewHolder viewHolder,T data); //需要重写的abstract函数convert
    @Override
    public int getItemCount() { return mDatas.size(); }
    public void removeItem(int position) //删除该位置的数据
    {
        mDatas.remove(position);
        notifyItemRemoved(position);
    }

    public interface OnItemClickListener
    {
        void onClick(int position);
        void onLongClick(int position);
    }

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



### ③、Goods 类：用于存放商品的各种信息。

```
public class Goods implements Serializable {
    String name; // 名字
    String price; // 价格
    String type; // 类型, 例如产地、重量
    String info; // 相应的参数
    String goods_firstletter; // 名字的首字母
    int imageid; // 对应的图片
    boolean markShopcar; // 是否存在于购物车
    boolean markStar; // 是否被收藏
    int index; // 每个商品唯一的下标, 用于区分
    boolean markRemove; // 判断是否要移除
    public Goods(String name1, String price1, String type1, String info1, String goods_firstletter1, int imageid1, int i)
    { // 构造函数
        name=name1;
        price=price1;
        type=type1;
        info=info1;
        goods_firstletter=goods_firstletter1;
        imageid=imageid1;
        markShopcar=false;
        markStar=false;
        index=i;
        markRemove=false;
    }

    public void setMarkShopcar() { markShopcar=true; } // 设置购物车
    public void resetMarkShopcar() { markShopcar=false; } // 取消购物车
    public void setMarkStar() { markStar=true; } // 设置收藏
    public void resetMarkStar() { markStar=false; } // 取消收藏
    public void setMarkRemove() { markRemove=true; }
    public void resetMarkRemove() { markRemove=false; }

    public String getName() { return name; } // 获取名字
    public String getPrice() { return price; } // 获取价格
    public String getType() { return type; } // 获取类型
    public String getInfo() { return info; } // 获取相应的参数
    public String getGoods_firstletter() { return goods_firstletter; } // 获取首字母
    public int getImageid() { return imageid; } // 获取图片id
    public int getIndex() { return index; } // 获取id
    public boolean getMarkShopcar() { return markShopcar; } // 获取购物车的状态
    public boolean getMarkStar() { return markStar; } // 获取收藏的状态
    public boolean getMarkRemove() { return markRemove; }
```

### ④、MainActivity 类：用于实现商品列表和购物车还有浮动按钮。

```
/*--- 商品信息-----*/
private String[] goodsName = new String[]
{
    "Enchated Forest", "Arla Milk", "Devondale Milk", "Kindle Oasis",
    "waitrose 早餐麦片", "McVities's 饼干", "Ferrero Rocher", "Maltesers",
    "Lindt", "Borggreve";
};
private int[] imageid = new int[]{R.mipmap.enchatedforest, R.mipmap.arla, R.mipmap.devondale, R.mipmap.kindle, R.mipmap.waitrose,
R.mipmap.maltesers, R.mipmap.ferrero, R.mipmap.maltesers, R.mipmap.lindt, R.mipmap.borggreve};

private String[] goodsName_Firstletter = new String[]
{
    "E", "A", "D", "K",
    "W", "M", "F", "M",
    "L", "B";
};
private String[] goodsPrice = new String[]
{
    "¥5.00", "¥59.00", "¥79.00", "¥2399.00",
    "¥179.00", "¥14.90", "¥132.59", "¥141.43", "¥139.43", "¥28.90";
};
private String[] goodsType = new String[]
{
    "作者", "产地", "产地", "版本", "重量", "产地", "重量", "重量", "重量", "重量", "重量"
};
private String[] goodsInfo = new String[]
{
    "Johanna Basford", "德国", "澳大利亚", "8GB", "2Kg", "英国", "300g", "118g", "249g",
    "640g"
};
```

#### ④、MainActivity 类：用于实现商品列表和购物车还有浮动按钮。（接上）

```
/*数据的准备*/
RecyclerView mRecyclerView;//商品列表的RecyclerView
List<Goods> goodsDetail;//储存商品信息的一个列表
List<Map<String,Object>> goodsList;//商品列表
List<Map<String,Object>> shoppingcarList;//购物车列表
ListView mListView;//购物车列表的ListView
boolean exchange;//判断当前为购物车还是主页
FloatingActionButton floatingActionButton;//浮动按钮
CommonAdapter commonAdapter;//RecyclerView使用的commonAdapter
SimpleAdapter simpleAdapter;//ListView使用的simpleAdapter

public void initial()//初始化
{
    mRecyclerView=(RecyclerView)findViewById(R.id.mrecyclerview);
    mRecyclerView.setVisibility(View.VISIBLE);//显示商品列表
    goodsDetail=new ArrayList<>();
    floatingActionButton=(FloatingActionButton)findViewById(R.id.shoppingcar);
    goodsList=new ArrayList<>();
    shoppingcarList=new ArrayList<>();
    mListView=(ListView)findViewById(R.id.shoppinglist);
    mListView.setVisibility(View.INVISIBLE);//不显示购物车列表
    exchange=false;
}

/*goods detail*/
/*分别向商品列表和购物车列表中加入初始化数据*/
for(int i=0;i<10;i++)
{
    Map<String,Object> temp=new HashMap<>();
    temp.put("goods_Firstletter",goodsName_Firstletter[i]);
    temp.put("goodsname",goodsName[i]);
    temp.put("index",i);
    goodsList.add(temp);
}
Map<String,Object> temp1=new HashMap<>();
temp1.put("goods_Firstletter","*");
temp1.put("goodsname","购物车");
temp1.put("price","价格");
shoppingcarList.add(temp1);

/*goods information*/
/*往goodsDetail里面添加Goods类型的数据*/
for(int i=0;i<10;i++)
{
    goodsDetail.add(new Goods(goodsName[i],goodsPrice[i],goodsType[i],goodsInfo[i],goodsName_Firstletter[i],imageid[i],i));
}

/*Recyclerview goodsList*/
mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
commonAdapter=new CommonAdapter<Map<String,Object>>(this,R.layout.shoppinglist_item,goodsList) {
    @Override
    /*viewholder */
    public void convert(ViewHolder viewHolder, Map<String, Object> data) {
        TextView first=viewHolder.getView(R.id.goodsFirst_Letter);
        first.setText(data.get("goods_Firstletter").toString());
        TextView name=viewHolder.getView(R.id.goodsName);
        name.setText(data.get("goodsname").toString());
    }
};
commonAdapter.setOnItemClickListener(new CommonAdapter.OnItemClickListener() {
    @Override
    public void onClick(int position)
    {
        /*设置点击后进入详情界面*/
        Intent goodlist2details=new Intent(MainActivity.this,GoodDetail.class);
        Bundle bundle1=new Bundle();
        int i=Integer.parseInt(goodsList.get(position).get("index").toString());
        bundle1.putSerializable("goods", goodsDetail.get(i));
        goodlist2details.putExtras(bundle1);
        startActivityForResult(goodlist2details,1);
    }
    @Override
    public void onLongClick(int position)
    {
        /*长按删除*/
        commonAdapter.removeItem(position);
        Toast.makeText(getApplicationContext(),"移除第"+position+"个商品",Toast.LENGTH_SHORT).show();
    }
});
mRecyclerView.setAdapter(commonAdapter);//为RecyclerView设置Adapter
```

#### ④、MainActivity 类：用于实现商品列表和购物车还有浮动按钮。（接上）

```
/*购物车*/
simpleAdapter=new SimpleAdapter(this,shoppingcarList,R.layout.shoppinglist_item,
    new String[] {"goods_Firstletter","goodsname","price"},new int[] {R.id.goodsFirst_Letter,R.id.goodsName,R.id.price});
mListView.setAdapter(simpleAdapter);
/*----按一下跳转-----*/
mListView.setOnItemClickListener((parent, view, position, id) → {
    if(position>0)
    {
        Intent shoppingcar2details=new Intent(MainActivity.this,GoodDetail.class);
        Bundle bundle2=new Bundle();
        int i=Integer.parseInt(shoppingcarList.get(position).get("index").toString());
        bundle2.putSerializable("goods",goodsDetail.get(i));
        shoppingcar2details.putExtras(bundle2);
        startActivityForResult(shoppingcar2details,1);
    }
});
/*----长按-----*/
mListView.setOnItemLongClickListener((parent, view, position, id) → {
    if(position>0)
    {
        String tempString=shoppingcarList.get(position).get("goodsname").toString();

        final AlertDialog.Builder alert=new AlertDialog.Builder(MainActivity.this);
        alert.setTitle("移除商品");
        alert.setMessage("从购物车中移除" + tempString + "?");
        alert.setPositiveButton("确定", (dialog, which) → {

            shoppingcarList.remove(position);

            simpleAdapter.notifyDataSetChanged();

        })

        .setNegativeButton("取消", (dialog, which) → {
            /*nothing happened*/
            dialog.cancel();
        })
        .create()
        .show();
    }
    return true;
});
/*----floatingActionButton悬浮按钮，切换购物车和商品列表界面-----*/
floatingActionButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
    if(exchange==false)//显示购物车列表
    {
        mListView.setVisibility(View.VISIBLE);
        mRecyclerView.setVisibility(View.INVISIBLE);
        floatingActionButton.setImageResource(R.mipmap.shoplist);
        exchange=!exchange;
    }
    else//显示商品列表
    {
        mListView.setVisibility(View.INVISIBLE);
        mRecyclerView.setVisibility(View.VISIBLE);
        floatingActionButton.setImageResource(R.mipmap.mainpage);
        exchange=!exchange;
    }
}
});
```

#### ④、MainActivity 类：用于实现商品列表和购物车还有浮动按钮。（接上）

```
@Override
protected void onActivityResult(int requestCode,int resultCode,Intent intent)
{
    //得到新打开商品详情关闭后返回的数据
    //requestCode提供给onActivityResult, 是以便确认返回的数据是从哪个Activity返回的。
    //这个requestCode和startActivityForResult中的requestCode相对应。
    //这整数resultCode是由子Activity通过其setResult()方法返回。
    if(requestCode==1&&resultCode==2)
    {
        Bundle bundle=intent.getExtras();
        if(bundle!=null)
        {
            Goods goods=(Goods)bundle.get("shoppingcar");
            if(goods != null)
            {
                if(goods.getMarkShopcar()==true)
                {
                    Map<String,Object> temp=new HashMap<>();
                    temp.put("goods_Firstletter",goods.getGoods_firstletter());
                    temp.put("goodsname",goods.getName());
                    temp.put("price",goods.getPrice());
                    temp.put("index",goods.getIndex());
                    shoppingcarList.add(temp);
                    simpleAdapter.notifyDataSetChanged();
                }
                if(goods.getMarkRemove()==true)
                {
                    commonAdapter.removeItem(goods.getIndex());
                }
            }
        }
    }
}
```

#### ⑤、GoodDetail 类，商品详情的设计。

```
ImageView goodsimage;// 图片
TextView detailname;// 名字
ImageView star;// 收藏星星
ImageView backbutton;// 返回按钮
TextView detailprice;// 价格
TextView detailtype;// 类型
TextView detailinfo;// 相应的参数
ImageView detail_shoppingcar;// 购物车按钮
TextView detailmoredetails;// 更多
ListView mListview;// 列表用于显示"一键下单","分享商品","不感兴趣","查看更多商品信息"
SimpleAdapter simpleAdapter;
List<Map<String,Object>> othersList;
Goods goods;
String others[]=new String[]{"一键下单","分享商品","不感兴趣","查看更多商品信息",""};

Bundle data_back =new Bundle();
protected void initial()// 初始化
{
    goodsimage=(ImageView)findViewById(R.id.goodsImage);
    detailname=(TextView)findViewById(R.id.detail_name);
    star=(ImageView)findViewById(R.id.star);
    backbutton=(ImageView)findViewById(R.id.backbutton);
    detailprice=(TextView)findViewById(R.id.detail_price);
    detailtype=(TextView)findViewById(R.id.detail_type);
    detailinfo=(TextView)findViewById(R.id.detail_info);
    detail_shoppingcar=(ImageView)findViewById(R.id.detail_shoppingcar);
    detailmoredetails=(TextView)findViewById(R.id.detail_moredetails);
    mListview=(ListView)findViewById(R.id.othersList);
    othersList=new ArrayList<>();
}
```

```

Bundle tmp = getIntent().getExtras();
// 获取从商品列表中发送的数据
if(tmp != null) goods = (Goods) tmp.get("goods");
goods.resetMarkShopcar(); // 初始化为未加入购物车
goods.resetMarkStar(); // 初始化为未加入收藏

if(goods!=null)
{
    detailname.setText(goods.getName());
    detailtype.setText(goods.getType());
    detailinfo.setText(goods.getInfo());
    detailprice.setText(goods.getPrice());
    goodsimage.setImageResource(goods.getImageid());
}

/*star 设置星星被点击的动作设定*/
star.setOnClickListener((v) -> {
    if(goods.getMarkStar()==true)
    {
        goods.resetMarkStar();
        star.setImageResource(R.mipmap.empty_star);
    }
    else
    {
        goods.setMarkStar();
        star.setImageResource(R.mipmap.full_star);
    }
});

/*shoppingcar 购物车的动作设定*/
detail_shoppingcar.setClickable(true);
detail_shoppingcar.setOnClickListener((v) -> {
    data_back.putSerializable("shoppingcar", goods);
    goods.setMarkShopcar();
    Toast.makeText(getApplicationContext(), "商品已添加到购物车", Toast.LENGTH_SHORT).show();
});

/*back 返回操作*/
backbutton.setOnClickListener((v) -> {
    Intent detail2goodlist=new Intent();
    detail2goodlist.putExtra("shoppingcar", goods);
    setResult(2, detail2goodlist);
    finish();
});

/*List 的设置，放置"一键下单"，"分享商品"，"不感兴趣"，"查看更多商品信息"*/
for(int i=0;i<others.length;i++)
{
    Map<String, Object> temp = new HashMap<>();
    temp.put("others", others[i]);
    othersList.add(temp);
}
simpleAdapter=new SimpleAdapter(this,othersList,R.layout.goods_details_other,
    new String[]{"others"},new int[]{R.id.others});
mListView.setAdapter(simpleAdapter);
mListView.setOnItemClickListener((parent, view, position, id) -> {
    (具体操作看课后实验内容这里不展示那部分了)
@Override
public boolean onKeyDown(int KeyCode, KeyEvent keyEvent)
{
    if(KeyCode==KeyEvent.KEYCODE_BACK)
    {
        Intent intent=new Intent(GoodDetail.this,MainActivity.class);
        //intent.putExtra("shoppingcar", goods);
        intent.putExtras(data_back);
        setResult(2,intent);
        finish();
    }
    return true;
}
}

```

### （3）实验遇到困难以及解决思路

在样式设计的时候，想借助 LinearLayout 来实现布局商品列表项，然而在实现的过程中发现其列表项变成了垂直排列，在查询了很久原因之后发现是由于默认是垂直排列，需要改成水平排列才可以。其次在按 FloatingActionButton 的按键切换逻辑问题，而且一开始进入界面后 RecyclerView 和 ListView 同时显示，后面修改了 FloatingActionButton 的逻辑和设置了初始化 ListView 不显示之后得以解决。然后是出现了一个很奇怪的 BUG，在长按删除了某个商品之后，点击原来该商品所在位置的其他商品，进入的详情仍为删除的商品，后面发现问题在于你选取的位置仍在列表中的序号是不变的，然后通过给 Goods 商品类加入了一个 index 值，来保证通过屏幕来获取其原位置。

## 四、 课后实验结果

为商品详情里面的四个列表项添加了各自的点击功能：

一键下单
分享商品
不感兴趣
查看更多商品信息

1、一键下单功能：实现在点击之后将商品添加到购物车并返回到商品列表界面。



```
if(position==0)
{
    data_back.putSerializable("shoppingcar",goods);
    goods.setMarkShopcar();
    Toast.makeText(getApplicationContext(),"商品已添加到购物车",Toast.LENGTH_SHORT).show();
    Intent detail2goodlist=new Intent();
    detail2goodlist.putExtra("shoppingcar",goods);
    setResult(2,detail2goodlist);
    finish();
}
```

2、分享商品：由于分享功能在查询后较为复杂，时间关系故暂未实现。





3、不感兴趣：将其从商品列表中删除，并返回商品列表。



4、查看更多商品信息：直接返回商品列表。

```
if(position==3)
{
    Intent detail2goodlist=new Intent();
    detail2goodlist.putExtra("shoppingcar",goods);
    setResult(2,detail2goodlist);
    finish();
}
```

## 五、 实验思考及感想

这次实验感觉学到了很多，比起上一个实验的难度感觉增加不少，最难的就是对于整个实验的设计了，虽然实验文档写的比较详细，但是在自己实验的时候一开始会有一种无从下手的感觉，然后通过找找网上的思路和询问下大佬的看法，总算有了自己的想法。我觉得如果能够设计好结构，那么实现起来会简单一点，例如我这里是分成了 4 个 XML 文件，2 个布局，2 个 item，5 个 JAVA 文件，其中最重要的两个是实现商品、购物车列表界面和商品详情的功能。而在布局好大概的结构的时候，再通过各个功能来分布实现功能，通过完成每一个小目标从而实现一个大目标。而在实际的设计中，还是有很多值得令人认真思考的部分的，例如 bundle 代码，可以借助 bundle 来实现两个 Activity 之间跳转的数据传输，例如在商品列表中，通过 bundle 将某个商品与关键字“goods”相匹配，在跳转到商品详情的时候，通过获取该关键字的内容即可得到在商品列表传入的数据。在课后设计方面，由于发现老师给的要求里面，虽然详情界面里面有 4 个列表项，但是并没有为其设计动作，为了软件的完整性，我就自己做了一些扩展，虽然都是和这次实验相同的功能，但是通过组合的方式，能给用户更好的体验，同时也能够通过这样的方式来让自己更加熟悉 Android。