# 南通大学计算机学院

## 《Java 软件实践》课程设计

# 报　告　书

设计题目 淘宝购物手机 APP 软件设计与开发

专业班级　软件工程 154

学生姓名　袁依吉

学　　号　1513032109

指导教师　陆培军

日　　期　2018/7/6

# 目　　录

# 1. 课程设计题目

淘宝购物手机 APP 软件设计与开发

# 2. 课程设计目的

通过本项目，学生可以掌握 Android 系统系架构；掌握 Android 系统上 APP 软件开发方法；了解服务器与客户端模式软件运行机制，掌握 APP 与服务器通过 JSON 格式进行数据交互的技术使用与软件开发技术；通过本项目学生可以对 Java 软件的整个开发平台、J2EE 技术、Android 技术等 Java 模块知识有一个系统全面的掌握，同时了解现在主流的软件设计思路。

# 3. 课程设计要求

(1)具体要求详见附件《淘宝购物手机 APP 软件设计与开发任务书》

(2)数据来源:

A、通过关键词查询，获取产品列表的 URL

http://s.m.taobao.com/search?event_submit_do_new_search_auction=1&_input_charset=utf-8&searchfrom=1&action=home%3Aredirect_app_action&from=1&q=%E8%BF%90%E5%8A%A8%E9%9E%8B&sst=1&n=20&buying=buyitnow&m=api4h5&wlsort=10&page=1

http://s.m.taobao.com/search?event_submit_do_new_search_auction=1&_input_charset=utf-8&searchfrom=1&action=home%3Aredirect_app_action&from=1&q={0}&sst=1&n=20&buying=buyitnow&m=api4h5&wlsort=10&page=1

B、通过产品 Numid 获取产品详情的 URL

https://acs.m.taobao.com/h5/mtop.taobao.detail.getdetail/6.0/?data=%7B%22itemNumId%22%3A%2210031645140%22%7D&qq-pf-to=pcqq.group

https://acs.m.taobao.com/h5/mtop.taobao.detail.getdetail/6.0/?data=%7B%22itemNumId%22%3A%22{0}%22%7D&qq-pf-to=pcqq.group

# 4. 课程设计报告内容

## 4.1 系统主要功能设计

### 4.1.1 功能描述

界面 1(首页界面):

APP 首次启动时，上方为 EditText，用户可以此输入搜索词，下方显示:

(1)如果没有收藏产品，则默认系统预设的产品，要求预设 12 个。

(2)如果有收藏产品显示以方块形式显示收藏的产品(图片、产品名称、价格、销量)

用户点击产品信息跳转到界面 3。用户输入搜索产品，单击“搜索”跳转到界面 2

界面 2(产品列表界面):

显示查询产品列表。产品列表中至少需要显示产品图片、产品标题、产品价格、产品销量，可以显示更多关于产品的信息。但至少需显示上述 4 项。列表以行显示，一行显示一个产品，用户单击某一行，则 APP 跳转显示到界面 3。

显示产品要求：要对直接从 API 接口获取的数据进行预处理，数据清洗，要求对于每个搜索词显示销量最多的 50 个产品,对于一屏显示不下的,必须可以上下滑动显示。

例如：用户输入“华为手机”，单击搜索后，需要从 API 接口获取 500 个华为手机的产品，在这些产品中也会许包含“华为手机壳”的产品，要想办法过滤掉，可以按照价格来过滤，手机的价格肯定比手机壳的价格要高很多。过滤不符合要求的产品，按照销量取前 50 名的产品显示在 APP 中。

界面 3(产品详情界面):

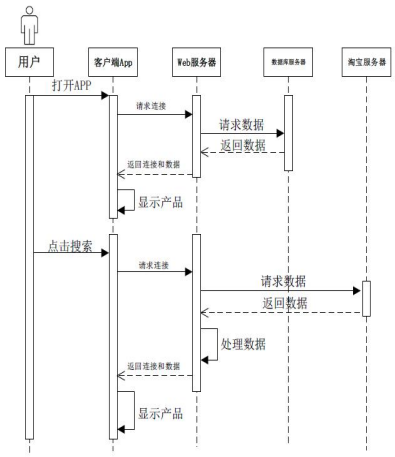
显示从界面跳转过来的产品的详细信息，至少需要包括：产品图片(能获取到的所有不同图片)，标题，价格，折扣价，销量，卖家昵称,其它信息可自行添加)，在界面的最下方显示“购买”、"收藏"、"查看"按钮，用户单击"购买"按钮， 跳转到淘宝对应的产品详情页，如果用户单击"收藏"，将该产品信息收藏(或以设计一张数据表保存收藏的产品)，如果该产品未被收藏过，则显示弹出信息" 收藏成功"，如果产品已被收藏过，则显示"收藏失败，该产品已被收藏，请单击查看按钮"。

界面 4(收藏产品管理界面):

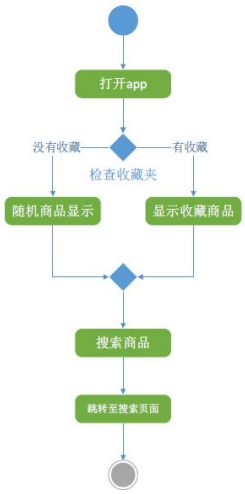
用户单击查看按钮后的显示已收藏产品界面，显示该用户所收藏的所有产品， 以列表形式显示

### 4.1.2 功能设计

时序图:

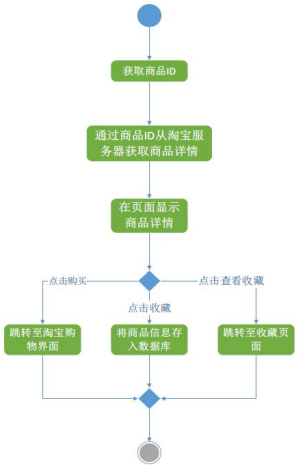



主页面活动图：

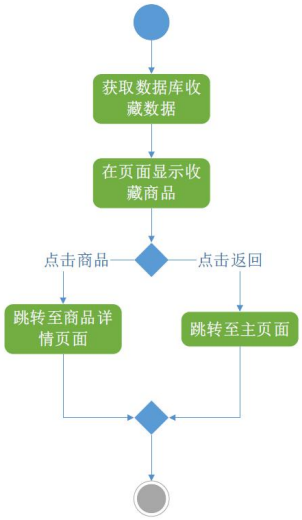



搜索界面活动图：


商品界面活动图：




收藏界面活动图：




## 4.2 系统设计与文件清单

java.com.example.taobao:

collection.java:收藏夹代码

details.java:产品详情的代码

list:产品列表的代码

MainActivity.java:主页面代码

MyData.java:数据库创建存储代码

assets:

litepal.xml:litepal 数据库框架

res.layout:

activity_main.xml:主页面布局

collection.xml:收藏夹布局

details.xml:物品详情布局

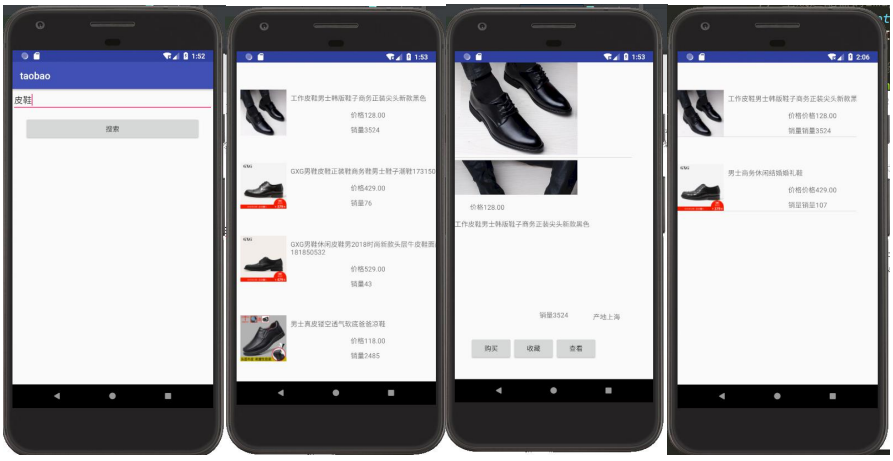
details_item.xml:物品详情 ListView 项目布局

item.xml:主页 GridView 项目布局

list.xml:列表 ListView 布局

list_item.xml:列表 GridView 项目布局、收藏 ListView 项目布局

## 4.3 系统代码实现




```
// 生成适配器的 ImageItem <====> 动态数组的元素，两者一一对应
ArrayList<HashMap<String, Object>> lstImageItem = new ArrayList<HashMap<String, Object>>();
        for (int i = 0; i < 12; i++) {
            int j=i+1;
            MyData goods = DataSupport.find(MyData.class , j);
            HashMap<String, Object> map = new HashMap<String, Object>();
            //map.put("ItemImage", goods.getImage());// 添加图像资源的 ID
            //map.put("ItemText", "NO." + goods.getTitle());// 按序号做 ItemText
            //map.put("ItemText2", "subNO." + goods.getPrice());
            lstImageItem.add(map);
        }

        // 生成适配器的 ImageItem <====> 动态数组的元素，两者一一对应
        SimpleAdapter saImageItems = new SimpleAdapter(this,
                lstImageItem,// 数据来源
                R.layout.item,// item 的 XML 实现
                // 动态数组与 ImageItem 对应的子项
                new String[] { "ItemImage", "ItemText" ,"ItemText2"},
                // Item 的 XML 文件里面的一个 ImageView,两个 TextView ID
                new int[] { R.id.imageView, R.id.textView,R.id.textView2 });
        // 添加并且显示
        gridView.setAdapter(saImageItems);
```

```
// 添加消息处理
        // 注册监听事件
        gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
                                    long arg3) {
                // TODO Auto-generated method stub
//                  Toast.makeText(MainActivity.this,
//                          arg2+"号", Toast.LENGTH_SHORT).show();
            }
        });


        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
//                  List<MyData> myDatas = DataSupport.findAll(MyData.class);
//                  DataSupport.deleteAll(MyData.class);
//                  Log.d(TAG, myDatas.toString());
                    try {
                        Log.d("tmpUrl","line92");
                        keyEdit = URLEncoder.encode(edit.getText().toString(), "utf-8");
                        if (!keyEdit.equals("")) {
                            tmpUrl=url.replace("{0}", keyEdit);
                            Intent intent=new Intent(MainActivity.this, list.class);
                            intent.putExtra("edit",tmpUrl);
                            startActivity(intent);
                        }

                    } catch (Exception e) {
                        e.printStackTrace();
                    }
//                  Intent intent=new Intent(MainActivity.this, list.class);
//                  intent.putExtra("edit",tmpUrl);
//                  startActivity(intent);
            }
        });

public void con() {
        new Thread(new Runnable() {
            @Override
            public void run() {
                //获取网址
```

```
                    Intent intent = getIntent();
                    intentUrl = intent.getStringExtra("edit");
                    //Log.d("list", "获取网址" + intentUrl);
                    //用 HttpClient 发送请求，分为五步
                    //第一步：创建 HttpClient 对象
                    HttpClient httpCient = new DefaultHttpClient();
                    //第二步：创建代表请求的对象,参数是访问的服务器地址
                    HttpGet httpGet = new HttpGet(intentUrl.toString());

                    try {
                        //第三步：执行请求，获取服务器发还的相应对象
                        HttpResponse httpResponse = httpCient.execute(httpGet);
                        //第四步：检查相应的状态是否正常：检查状态码的值是 200 表示正常
                        if (httpResponse.getStatusLine().getStatusCode() == 200) {
                            //第五步：从相应对象当中取出数据，放到 entity 当中
                            HttpEntity entity = httpResponse.getEntity();
                            String response = EntityUtils.toString(entity,"utf-8");//将 entity 当中的
数据转换为字符串

                            //在子线程中将 Message 对象发出去
                            Message message = new Message();
                            message.what = 123;
                            message.obj = response.toString();
                            handler.sendMessage(message);
                        }

                    } catch (Exception e) {
                        e.printStackTrace();
                    }



                }
            }).start();//这个 start()方法不要忘记了
        }



        Handler handler = new Handler() {
            public void handleMessage(Message msg) {
                super.handleMessage(msg);
                if (msg.what == 123) {
                    htmlContent = (String) msg.obj;
                    tran(htmlContent);
```

```
            }
        }
    };


        //获取网页源代码转换成 JSON

    public void tran(String htmlContent) {

        try {

            Log.d("list", "转换前" + htmlContent);
            JSONObject obj = new JSONObject(htmlContent);

            JSONArray itemData = obj.getJSONArray("listItem");

            final ArrayList<HashMap<String, Object>> lstImageItem = new
ArrayList<HashMap<String, Object>>();//新建 ArrayList

            for (int i = 0; i < itemData.length(); i++) {

                JSONObject item = itemData.getJSONObject(i);

                String imgurl = item.getString("pic_path").replace("60x60", "100x100");
                String title = item.getString("title");
                String price = item.getString("price");
                String sold = item.getString("sold");
                String item_id = item.getString("item_id");
                String area = item.getString("area");
                //Log.d(TAG, String.valueOf(imgurl));
                //                  Drawable drawable = loadImageFromNetwork(imgurl);
                //                  imageView2.setImageDrawable(drawable) ;
                Bitmap bitImg = getHttpBitmap(imgurl);

                HashMap<String, Object> map = new HashMap<String, Object>();

                //Log.d(TAG, String.valueOf(bitImg));
                map.put("ItemImage", bitImg);// 添加图像资源的 ID
                map.put("ItemImgurl", imgurl);
                map.put("ItemTitle", title);// 按序号做 ItemText
                map.put("ItemPrice", "价格" + price);
                map.put("ItemSold", "销量" + sold);
                map.put("ItemFrom", "产地" + area);
```

```
                map.put("ItemId", item_id);
                lstImageItem.add(map);

            }//for 循环

            // 生成适配器的 ImageItem <====> 动态数组的元素，两者一一对应
            final SimpleAdapter saImageItems = new SimpleAdapter(this,
                    lstImageItem,// 数据来源
                    R.layout.list_item,// item 的 XML 实现
                    // 动态数组与 ImageItem 对应的子项
                    new String[]{ "ItemImage","ItemTitle", "ItemPrice", "ItemSold"},
                    // Item 的 XML 文件里面的一个 ImageView,两个 TextView ID
                    new int[]{ R.id.imageView2, R.id.textView3, R.id.textView4,
R.id.textView5});

    //适配器图片显示
            saImageItems.setViewBinder(new SimpleAdapter.ViewBinder() {

                @Override
                public boolean setViewValue(View view, Object data, String textRepresentation) {
                    // TODO Auto-generated method stub
                    if (view instanceof ImageView && data instanceof Bitmap) {

                        ImageView iv = (ImageView) view;
                        iv.setImageBitmap((Bitmap) data);
                        return true;
                    } else
                        return false;
                }
            });

            // 添加并且显示
            //gridView = (GridView) findViewById(R.id.GridView2);
    //列表网格显示
            gridView.setAdapter(saImageItems);

            gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
                @Override
                public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
                    HashMap<String, Object> map = (HashMap<String,
Object>)lstImageItem.get(i);
                    String title = (String)map.get("ItemTitle");
                    String price = (String)map.get("ItemPrice");
                    String sold = (String)map.get("ItemSold");
```

```
                String from = (String)map.get("ItemFrom");
                String id = (String)map.get("ItemId");
                String imgUrl = (String)map.get("ItemImgurl");

                Intent intent = new Intent(list.this, details.class);
                intent.putExtra("title",title);
                intent.putExtra("price",price);
                intent.putExtra("sold",sold);
                intent.putExtra("from",from);
                intent.putExtra("id",id);
                intent.putExtra("ItemImgurl",imgUrl);
                startActivity(intent);
            }
        });
    } catch (Exception e) {
        e.printStackTrace();
    }
}

//地址转 Bitmap
    public static Bitmap getHttpBitmap(final String url){

        URL myFileUrl = null;
        Bitmap bitmap = null;
        try {
            myFileUrl = new URL(url);
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        try {
            HttpURLConnection conn = (HttpURLConnection) myFileUrl
                    .openConnection();
            conn.setDoInput(true);
//              conn.connect();
            InputStream is = conn.getInputStream();
            bitmap = BitmapFactory.decodeStream(is);
            is.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return bitmap;
    }
//DB 创建存储
public MyData(String image, String price, String sold, String title, String location, String item_id) {
```

```
        this.image = image;
        this.price = price;
        this.sold = sold;
        this.title = title;
        this.location = location;
        this.item_id = item_id;
    }
//get()略 set()略
```

## 4.4 系统测试

### 4.4.1 系统测试数据清单

| 用例编号 | 用例操作 | 数据 | 预期结果 |
|---|---|---|---|
| 1 | 在输入框输入数据搜索 | 皮鞋 | 皮鞋搜索界面 |
| 2 | 在输入框输入数据搜索 | null | 随即搜索界面 |
| 3 | 在主页面点击商品 | | 进入详情界面 |
| 4 | 在搜索页面点击商品 | | 进入详情界面 |
| 5 | 点击购买按钮 | | 跳转至淘宝 |
| 6 | 点击收藏按钮（未收藏） | | 显示收藏成功 |
| 7 | 点击收藏按钮（已收藏） | | 显示收藏失败 |
| 8 | 点击查看按钮 | | 进入收藏界面 |
| 9 | 收藏夹界面点击商品 | | 进入详情界面 |
| 10 | 点击返回按钮 | | 返回主界面 |

### 4.4.2 系统测试结果

| 用例编号 | 实际结果 | 是否符合期望（Y/N） |
|---|---|---|
| 1 | 皮鞋搜索界面 | Y |
| 2 | 无 | N |
| 3 | 无 | N |
| 4 | 进入详情界面 | Y |
| 5 | 跳转至淘宝 | Y |
| 6 | 显示收藏成功 | Y |
| 7 | 显示收藏失败 | Y |

| | | |
|---|---|---|
| 8 | 进入收藏界面 | Y |
| 9 | 无 | N |
| 10 | 无 | N |

## 4.5 系统存在的主要问题及注意事项

主页面，收藏页面无法进入详情

无法随即搜索

布局丑

## 4.6 设计总结及体会

对于这次课设，我学会了如何在一个人项目中包装不同的方法，如何寻求帮助解决遇到的问题，耦合不同的方法，通时也暴露了我的知识的不足，对代码的理解能力，逻辑思维能力，编程能力的不足，遇到问题不能退缩，通过不断的调试，使自己成长。

---

## 参考书目


1. UML 软件建模教程[M].卫红春
2. Android 应用程序开发（第 3 版）[M].王向灰，张国印，沈洁
3. 软件工程导论（第 6 版）[M].张海藩，牟永敏
4. 软件测试方法和技术（第 3 版）[M].朱少民


## 附件

《淘宝购物手机 APP 软件设计与开发任务书》