学　号	201825040233
成　绩	

移动终端开发技术

课程设计报告

题　　目	仿饿了么界面
班　　级	**18** 级软件技术二班
学　　号	**201825040233**
姓　　名	田向攀
小组成员	田向攀，郭李鹏，张茂鹏，司斌
指导教师	陈媛媛

2019 年 **7** 月 **3** 日

目录

1 引言

1.1 研究背景

近年来，随着 Internet 的迅速崛起，互联网已日益成为收集提供信息的最佳渠道并逐步进入传统的流通领域。

于是电子商务开始流行起来，越来越多的商家在网上建起在线商店，向消费者展示出一种新颖的购物理念。

随着网络购物的飞速发展，在淘宝、京东、唯品会、当当、亚马逊等网上商城的巨大影响力下，我国网络商城的前景有着非常广阔的发展空间。CNNIC 数据显示，2014 年我国手机网络购物用户规模达到 2.36 亿，增长率为 63.5%，是网络购物市场整体:用户规模增长速度的 3.2 倍，手机购物的使用比例提升了 13.5 个百分点达到 42.4%。CNNIC 研究显示，手机购物并非 PC 购物的替代，而是在移动环境下产生增量消费，并且重塑线下商业形态促成交易，从而推动网络购物移动化发展趋势。因此手机购物对于中国的经济有着重要的影响。同样，随着近年来 4G 网络技术的发展，手机网速也有了飞速提高，手机购物也必将成为未来网络购物的主流渠道。

在智能手机当中，苹果系统与安卓系统是两大主流系统，两大系统中又以安卓系统为主流，有研究表明:安卓系统将从 2014 年市场占有率的 49%，上升 10%，在 2015 年市场占有率接近 59%。因此《基于 Android 的网.上商店系统的设计与实现》这一课题有着非常的应用背景，较高的应用价值。

网上购物系统作为 B2B，B2C（BusinesstoCustomer,即企业对消费者），C2C（CustomertoCustomer，

即消费者对消费者)电子商务的前端商务平台，在其商务活动全过程中起着举足轻重的作用。本文对如何

开发出一个具有基本网上交易功能的 B2C 网上商城，给出了具体的指导。主要介绍基于 UML 的 B2C 网上商城

系统分析与设计的基本内容。对 B2C 网上商城的例图、顺序图、活动图、类图数据库设计和 jSP 编程等，都提出了具有针对性的解决方法。

1.2 网上商城系统描述

（1）投资少，回收快。一项针对中国中小企业的情况调查显示，个人在网下启动销售公司的平均费用至少 5 万元，而网上开店建店成本非常小。一般说，筹办一家网上的商店投入

很小，不用去办营业执照，不用去租门面，不用囤积货品，所需资金不过 1500 元左右；网上商店比同等规模的地面商店“租金”要低得多，同时租金不会因为营业面积的增加而增加，投资者也不用为延长营业时间而增加额外的费用。

（2）基本不需要占压资金。传统商店的进货资金少则几千元，多则数万元，而网上商店则不需要压资金。

（3）24 小时营业时间。网上商店延长了商店的营业时间，一天 24 小时、一年 365 天不停地运作，无须专人值班看店，都可照常营业。传统店铺的营业时间一般为 8-12 小时，遇上坏天气或者老板、店员有急事也不得不暂时休息。

（4）不受店面空间的限制。哪怕只是街边小店，在网上却可以拥有百货大楼那么大的店面，只要投资者愿意，可以摆上成千上万种商品。目前国内最大的专业拍卖网站同时在线的商品要超过 10 万件，已超过一些大超市。

（5）不受地理位置影响。不管客户离店有多远，也不管顾客是国内还是国外，在网上，客户一样可以很方便地找到并购买商品。这令消费群体突破了地域的限制，变得无限广阔了。

1.3 开发运行环境

Android 以 java 作为开发语言，jdk 是进行 java 开发时必需的开发包.Eclipse 是著名的跨平台的自由集成开发环境(IDE)。最初主要用来 Java 语言开发，通过安装不同的插件 Eclipse 可以支持不同的计算机语言，比如 C++和 Python 等开发工具。Eclipse 的本身只是一个框架平台，但是众多插件的支持使得 Eclipse 拥有其他功能相对固定的 IDE 软件很难具有的灵活性。在大量插件的”配合”下，Eclipse 完全可以满足从企业级 java 应用到手机终端 java 游戏的开发.Google 官方也提供了基于 Eclipse 的 Android 开发插件 ADT,所以本软件开发选择 Eclipse 作为开发 IDE.

1.4 相关技术简介

Vue 有自己的脚手架构建工具 vue-cli,使用起来非常方便，使用 webpack 来集成各种开发便捷工具，比如：Hot-reloadVue 的热更新，修改代码之后无需手动刷新网页，对前端开发来说非常方便 PostCss，再也不用去管兼容性的问题了，只针对 chrome 这样的现代浏览器写 css 代码，会自动编译生成兼容多款浏览器的 css 代码 ESlint，统一代码风格，规避低级错误，对于有代码洁癖的人来说是不可或缺的 Bable，ES2015 出来已经有一段时间了，但是不少浏览器还没有兼容 ES6，有了 bable，放心使用 ES6 语法，它会自动转义成 ES5 语法 SCSS，一款 CSS 预处理器，编译后成正常的 CSS 文件。为 CSS 增加一些编程的特性
除此之外，vue-cli 已经使用 node 配置了一套本地服务器和安装命令等，本地运行和打包只需要一个命令就可以搞定，非常的方便。

2 系统分析

2.1 系统的整体架构

app.vue

header.vue--头部组件

star.vue--星星评分组件

goods.vue--商品组件

shopcart.vue--购物车组件,包括小球飞入购物车动画

cartcontrol.vue--购买加减图标控件--选中数量返回给父组件 goods，goods 响应后，重新计算选中数量，将数据发送给购物车组件，

food.vue--商品详情页

ratingselect.vue--评价内容筛选组件

ratings.vue--评论组件

ratingselect.vue--评价内容筛选组件

seller.vue--商家组件

独立组件

split.vue--关于分割线组件

2.2 系统功能划分

Goods、Ratings、Seller 组件视图均可上下滚动

商品页点击左侧 menu，右侧 list 对应跳转到相应位置

点击 list 查看商品详情页，父子组件的通信

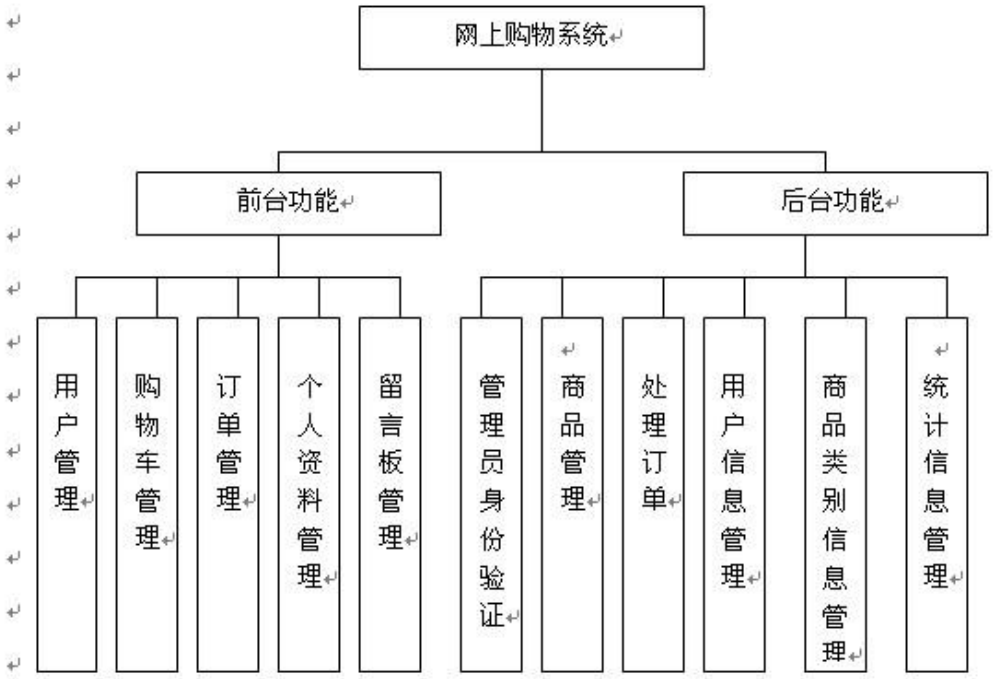
评论内容可以筛选查看

购物车组件，包括添加删除商品及动效，购物控件与购物车组件之间为兄弟组件通信，点击购物车图标，展示已选择的商品列表

商家实景图片可以左右滑动

loaclStorage 缓存商家信息（id、name）

3.前台与后台管理系统设计



4.列表悬停的设计

4.1 悬停设计的分析

现在做项目列表什么的基本抛弃了 ListView 改用 RecyclerView，上篇博客中的标题悬停也是使用了一个 RecyclerView 的开源项目 sticky-headers-recyclerview,不过写这个 demo 的时候遇到了两个坑

sticky-headers-recyclerview 做悬停标题的时候 scrollToPosition(intposition)方法滚动的位置不准确。

当布局复杂点的时候如果 RecyclerView 的宽度自适应或者使用权重百分比之类可能会导致 header 显示空白。

并且该开源项目作者已经停止维护，所以这次又换回了 StickyListHeadersListView。需要购物车 Demo 的很多都是新手，这里简单介绍下 StickyListHeadersListView 的使用

4.2 悬停设计的搭建

AS 引用 gradle 文件 dependencies 内添加

compile'se.emilsjolander:stickylistheaders:2.7.0'

xml 文件中使用 StickyListHeadersListView 代替 ListView

<se.emilsjolander.stickylistheaders.StickyListHeadersListView

android:layout_width="match_parent"

android:background="#fff"

android:id="@+id/itemListView"

android:layout_height="match_parent">

</se.emilsjolander.stickylistheaders.StickyListHeadersListView>

Adapter 继承 BaseAdapter 和接口 StickyListHeadersAdapter

StickyListHeadersAdapter 接口包括两个方法

ViewgetHeaderView(intposition,ViewconvertView,ViewGroupparent);

longgetHeaderId(intposition);

代码中使用和 ListView 一样，下面是几个特有的方法,看方法名也很容易理解用途

5 左右列表联动

5.1 左右列表联动的分析

联动主要有两个效果

-左侧列表点击选择分类，右侧列表滑动到对应分类

-右侧列表滑动过程中左侧列表高亮的分类跟随变化

第一个效果简单，左侧列表 item 添加点击事件，事件中调用右侧列表的 setSelection(intpositon)方法。

第二个效果要给右侧列表添加 ScrollListener，根据列表中显示的第一条数据设置左侧选中的分类

5.2 左右列表联动的设计

listView.setOnScrollListener(newAbsListView.OnScrollListener(){

@Override

publicvoidonScrollStateChanged(AbsListViewview,intscrollState){

}

@Override

publicvoidonScroll(AbsListViewview,intfirstVisibleItem,intvisibleItemCount,i

```
nttotalItemCount){
    //根据 firstVisibleItem 获取分类 ID，根据分类 id 获取左侧要选中的位置
    GoodsItemitem=dataList.get(firstVisibleItem);
    if(typeAdapter.selectTypeId!=item.typeId){
    typeAdapter.selectTypeId=item.typeId;
    typeAdapter.notifyDataSetChanged();
    //左侧列表是个 RecyclerView 所以使用 smoothScrollToPosition(intposition)使当对应 position 的 item 可以滚动显示出来
    rvType.smoothScrollToPosition(intposition)(getSelectedGroupPosition(item.typeId));
    }
    }
    });
```

6 添加商品的动画

6.1 添加商品动画的分析

添加商品一共有三个动画
-当商品从 0 到 1 旋转左移显示出减号按钮
-当商品从 1 到 0 减号按钮旋转右移消失
-添加商品时抛物线动画添加到购物车图标

6.2 添加商品动画的设计

动画的代码设置方法

```
//显示减号的动画
privateAnimationgetShowAnimation(){
AnimationSetset=newAnimationSet(true);
RotateAnimationrotate=newRotateAnimation(0,720,RotateAnimation.RELATIVE_TO_SELF,0.5f,RotateAnimation.RELATIVE_TO_SELF,0.5f);
set.addAnimation(rotate);
TranslateAnimationtranslate=newTranslateAnimation(
TranslateAnimation.RELATIVE_TO_SELF,2f
```

```
,TranslateAnimation.RELATIVE_TO_SELF,0
,TranslateAnimation.RELATIVE_TO_SELF,0
,TranslateAnimation.RELATIVE_TO_SELF,0);
set.addAnimation(translate);
AlphaAnimationalpha=newAlphaAnimation(0,1);
set.addAnimation(alpha);
set.setDuration(500);
returnset;
}
//隐藏减号的动画
privateAnimationgetHiddenAnimation(){
AnimationSetset=newAnimationSet(true);
RotateAnimationrotate=newRotateAnimation(0,720,RotateAnimation.RELATIVE_TO_SELF,
0.5f,RotateAnimation.RELATIVE_TO_SELF,0.5f);
set.addAnimation(rotate);
TranslateAnimationtranslate=newTranslateAnimation(
TranslateAnimation.RELATIVE_TO_SELF,0
,TranslateAnimation.RELATIVE_TO_SELF,2f
,TranslateAnimation.RELATIVE_TO_SELF,0
,TranslateAnimation.RELATIVE_TO_SELF,0);
set.addAnimation(translate);
AlphaAnimationalpha=newAlphaAnimation(1,0);
set.addAnimation(alpha);
set.setDuration(500);
returnset;
}
//执行动画只需给对应控件 setAnimation 然后调用 setVisibility 方法即可
{
....
tvMinus.setAnimation(getHiddenAnimation());
tvMinus.setVisibility(View.GONE);
}
```

实现过程

Ⅰ.首先点击加号图标，拿到控件在屏幕上的绝对坐标，回调 activity 显示动画

```
int[]loc=newint[2];
v.getLocationInWindow(loc);
activity.playAnimation(loc);
```

Ⅱ创建动画的控件并添加到根部局并在动画结束后移除动画 view

```
publicvoidplayAnimation(int[]start_location){
ImageViewimg=newImageView(this);
img.setImageResource(R.drawable.button_add);
setAnim(img,start_location);
}
//创建动画平移动画直接传递偏移量
privateAnimationcreateAnim(intstartX,intstartY){
int[]des=newint[2];
imgCart.getLocationInWindow(des);
AnimationSetset=newAnimationSet(false);
AnimationtranslationX=newTranslateAnimation(0,des[0]-startX,0,0);
//线性插值器默认就是线性
translationX.setInterpolator(newLinearInterpolator());
AnimationtranslationY=newTranslateAnimation(0,0,0,des[1]-startY);
//设置加速插值器
translationY.setInterpolator(newAccelerateInterpolator());
Animationalpha=newAlphaAnimation(1,0.5f);
set.addAnimation(translationX);
set.addAnimation(translationY);
set.addAnimation(alpha);
set.setDuration(500);
returnset;
}
//计算动画 view 在根部局中的坐标添加到根部局中
privatevoidaddViewToAnimLayout(finalViewGroupvg,finalViewview,
int[]location){
```

```
intx=location[0];
inty=location[1];
int[]loc=newint[2];
vg.getLocationInWindow(loc);
view.setX(x);
view.setY(y-loc[1]);
vg.addView(view);
}
//设置动画结束移除动画 view
privatevoidsetAnim(finalViewv,int[]start_location){
addViewToAnimLayout(anim_mask_layout,v,start_location);
Animationset=createAnim(start_location[0],start_location[1]);
set.setAnimationListener(newAnimation.AnimationListener(){
@Override
publicvoidonAnimationStart(Animationanimation){
}
@Override
publicvoidonAnimationEnd(finalAnimationanimation){
//直接 remove 可能会因为界面仍在绘制中成而报错
mHanlder.postDelayed(newRunnable(){
@Override
publicvoidrun(){
anim_mask_layout.removeView(v);
}
},100);
}
@Override
publicvoidonAnimationRepeat(Animationanimation){
}
});
v.startAnimation(set);}
```

7 底部弹出购物车清单

7.1 购物清单的分析

集成简单，效果多样这里简单介绍一下使用方法

7.2 购物清单的设计

xml 中使用 BottomSheetLayout 包裹弹出 view 时候的背景布局，BottomSheetLayout 继承自帧布局

```
<com.flipboard.bottomsheet.BottomSheetLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/bottomSheetLayout"
android:layout_width="match_parent"
android:layout_height="match_parent">
<LinearLayout
android:orientation="horizontal"
android:layout_width="match_parent"
android:layout_height="match_parent">

<android.support.v7.widget.RecyclerView
android:layout_width="100dp"
android:id="@+id/typeRecyclerView"
android:layout_height="match_parent">
</android.support.v7.widget.RecyclerView>

<se.emilsjolander.stickylistheaders.StickyListHeadersListView
android:layout_width="match_parent"
android:background="#fff"
android:id="@+id/itemListView"
android:layout_height="match_parent">
</se.emilsjolander.stickylistheaders.StickyListHeadersListView>
</LinearLayout>
</com.flipboard.bottomsheet.BottomSheetLayout>
```

代码中使用很简单
//弹出 ViewbottomSheet 即是要弹出的 view
bottomSheetLayout.showWithSheetView(bottomSheet);

//代码隐藏 view(点击弹出 view 以外的地方可以隐藏弹出的 view，向下滑动也可以)
bottomSheetLayout.dismissSheet();

8 数据的同步

8.1 数据同步的分析

//商品列表
privateArrayList<GoodsItem>dataList;
//分类列表
privateArrayList<GoodsItem>typeList;
//已选择的商品
privateSparseArray<GoodsItem>selectedList;
//用于记录每个分组选择的数目
privateSparseIntArraygroupSelect;
SparseArray 这个类其实就是 HashMap<Integer,Object>

不过 SparseArray 既可以根据 key 查找 Value，也可以根据位置查找 value，性能比 HashMap 高，是官方推荐的替代类，
同样 SparseIntArray 其实是 HashMap<Integer,Integer>的替代者。

Activity 里实现了下面几个方法,用于数据统一管理
列表中显示的商品购买数量统一从 activity 获取,商品的加减统一调用 Activity 的方法然后 notifiDatasetChanged

8.2 数据同步的设计

/**

```
*Item 代表商品的购买数量加一
*@paramitem
*@paramrefreshGoodList 是否刷新商品 list
*/
publicvoidadd(GoodsItemitem,booleanrefreshGoodList){

intgroupCount=groupSelect.get(item.typeId);
if(groupCount==0){
groupSelect.append(item.typeId,1);
}else{
groupSelect.append(item.typeId,++groupCount);
}

GoodsItemtemp=selectedList.get(item.id);
if(temp==null){
item.count=1;
selectedList.append(item.id,item);
}else{
temp.count++;
}
update(refreshGoodList);
}
/**
*Item 商品的购买数量减一
*@paramitem
*@paramrefreshGoodList 是否刷新商品 list
*/
publicvoidremove(GoodsItemitem,booleanrefreshGoodList){

intgroupCount=groupSelect.get(item.typeId);
if(groupCount==1){
groupSelect.delete(item.typeId);
```

```
}elseif(groupCount>1){
groupSelect.append(item.typeId,--groupCount);
}

GoodsItemtemp=selectedList.get(item.id);
if(temp!=null){
if(temp.count<2){
selectedList.remove(item.id);
}else{
item.count--;
}
}
update(refreshGoodList);
}

/**
*刷新界面总价、购买数量等
*@paramrefreshGoodList 是否刷新商品 list
*/
privatevoidupdate(booleanrefreshGoodList){
...
}

//根据商品 id 获取当前商品的采购数量
publicintgetSelectedItemCountById(intid){
GoodsItemtemp=selectedList.get(id);
if(temp==null){
return0;
}
returntemp.count;
}
//根据类别 Id 获取属于当前类别的数量
```

```
publicintgetSelectedGroupCountByTypeId(inttypeId){
returngroupSelect.get(typeId);
}
```

9 开发过程中的问题及其总结

（1）better-scroll 插件在移动端使用时需要设置 click：true，否则移动端滑动无效

（2）分开设置 css 样式：

图标 icon.css--文字图标样式，通过 icommon.io 网站将 svg 图片转成文字图标样式

公共 base.css--处理设备像素比的一些样式，针对 border-1px 问题，不同设备像素比，显示的线条粗细不同

工具 mixin.css--设置 border-1px 样式和背景样式

（3）sticky-footer 布局

在 header 组件的详情页采用 sticky-footer 布局，主要特点是如果页面内容不够长的时候，页脚块粘贴在视窗底部；如果内容足够长时，页脚块会被内容向下推送

（4）要求自适应的布局

1、左侧宽度固定，右侧宽度自适应

2、元素宽度自适应设备宽度，且元素要求等宽高样式

（5）背景模糊效果

filter：blur(10px),注意，所有在内的子元素也会模糊，包括文字，所以采用定位布局，背景单独占用一个层，ios 有一个设置 backdrop-filter:blur(10px)，只会模糊背景,但不支持 android

（6）transition 过渡

在购买控件中使用 transition 过渡效果，实现添加减少按钮的动效，和小球飞入购物车的动效（模仿贝塞尔曲线的效果）

vue2.x 里面定义了 transition 过渡状态，

name-string,用于自动生成 CSS 过渡类名。

（7）seller 组件：

问题一：seller 页面中商品商家实景图片横向滚动

解决方案：每个 li 要 display：inline-block，因为 width 不会自动撑开父级 ul，所以需要将计算后的宽度赋值给 ul 的 width，（每一张图片的 width+margin）*图片数量-一个 margin，因为最后一张图片没有 margin

同时 newBScroll 里面要设置 scrollX:true,eventPassthrough:'vertical',//滚动方向横向

问题二：打开 seller 页面，无法滚动

问题分析：出现这种现象是因为 better-scroll 插件是严格基于 DOM 的，数据是采用异步传输的,页面刚打开,DOM 并没有被渲染,所以,要确保 DOM 渲染了,才能使用 better-scroll，解决方案：用到 mounted 钩子函数，同时必须搭配 this.$nextTick()

问题三：在 seller 页面，刷新后，无法滚动

问题分析：出现这种情况是因为 mounted 函数在整个生命周期中只会只行一次

解决方案：使用 watch 方法监控数据变化，并执行滚动函数

this._initScroll();this._initPicScroll();

（8）缓存数据

使用 window.localStorage 保存和设置缓存信息，封装在 store.js 文件内

（9）解析 url，得到商家信息，包括 id，name，在获取数据时，直接赋值，商家的 id 或 name 会被丢掉

我们需要将得到的 id 和 name 带到数据中，实际上在获取数据的时候，并没有带着 id 和 name，这时就要用到 es6 语法中 Object.assign(),官方解释为：可以把任意多个的源对象自身的可枚举属性拷贝给目标对象，然后返回目标对象。

（10）goods,ratings,seller 组件之间切换时会重新渲染

解决方案：在 app.vu 内使用 keep-alive，保留各组件状态，避免重新渲染

10 心得体会

终于还是结束了，为期半学年的 Android 学习止于这篇设计报告。通过这半学年的学学习我学到了很多，首先是对计算机新领域又有了了解和认知，当然更多的是对 Android 这门课程有了更深刻的认识。其实学了它才知道 Android 并不是很难，所以这次课程设计我们组竭尽其所学，尽量使课程设计能够包含所学知识，凸显更多的亮点。

通过这次课程设计，让我更加深刻了解了课本知识，和对以往知识的疏忽得以补充。除此之外更多的是学会了如何学习课堂之外的知识。这次课程设计用到了很多我们未知的知识，这就需要我们学会利用网络和和帮助文档。 课程设计是培训学生运用专业所学的理论知识和专业知识来分析解决现实问题的重要环节是对所学知识的复习和巩固。此次设计让我明白了一个深刻的道理：团队精神固然很重要，但人往往还是要靠自己的努力，自己亲身去经历，这样自己心里才会踏实，学到的东西才会更多。