**中国矿业大学计算机学院**

**2019级本科生课程报告**

课程名称 Android应用开发技术

报告时间 2021年12月21日

学生姓名 胡钧耀

学 号 06192081

专 业 计算机科学与技术

任课教师 徐东红

摘 要

为了巩固所学 Android 基础知识，要开发一款仿美团外卖的项目，该项目与我们平常看到的美团外卖项目界面比较类似，展示的内容包括店铺、菜单、购物车、订单与支付等信息。仿美团外卖项目是一个网上订餐项目，该项目中包含订餐的店铺、各店铺的菜单、购物车以及订单与付款等模块。在店铺列表中可以看到店铺的名称、月售数量、起送价格与配送费用、配送时间以及店铺特色等信息，点击店铺列表中的任意一个店铺，程序会进入到店铺详情界面，该界面主要用于显示店铺中的菜单信息，同时可以将想要吃的菜添加到购物车中，选完菜之后可以点击该界面中的“去结算”按钮，进入到订单界面，在该界面核对已点的菜单信息，并通过“去支付”按钮进行付款。为了增强软件的真实性与安全性，还加入了轮播广告栏 Banner 设计、欢迎界面、登录界面、注册界面，使用 MD5 进行用户名密码加密，利用 SharePreference 进行内部存储。

关键词：Android, 网上订餐, Banner 设计, MD5, SharePreference

项目演示：https://www.bilibili.com/video/BV1NL411L73Q



目 录

[1 课设目的与要求 1](#_Toc91716488)

[1.1 课设目的 1](#_Toc91716489)

[1.2 课设要求 1](#_Toc91716490)

[2 课设概述 1](#_Toc91716491)

[3 开发环境配置 1](#_Toc91716492)

[3.1 操作系统 1](#_Toc91716493)

[3.2 开发工具 1](#_Toc91716494)

[3.3 API版本 1](#_Toc91716495)

[3.4 主要的库和依赖版本 1](#_Toc91716496)

[4 所需环境与库安装准备过程 2](#_Toc91716497)

[5 功能实现 6](#_Toc91716498)

[5.1 店铺功能业务实现 6](#_Toc91716499)

[5.1.1 搭建标题栏布局 6](#_Toc91716500)

[5.1.2 搭建广告栏布局（轮播广告） 8](#_Toc91716501)

[5.1.3 搭建店铺页面布局 10](#_Toc91716502)

[5.1.4 搭建店铺列表条目界面布局 12](#_Toc91716503)

[5.1.5 封装店铺信息与菜品信息的实体类 14](#_Toc91716504)

[5.1.6 编写广告栏的适配器 19](#_Toc91716505)

[5.1.7 编写店铺列表适配器 20](#_Toc91716506)

[5.1.8 实现店铺界面显示功能 22](#_Toc91716507)

[5.2 店铺功能业务实现 24](#_Toc91716508)

[5.2.1 实现店铺界面显示功能 24](#_Toc91716509)

[5.2.2 搭建菜单列表条目界面布局 31](#_Toc91716510)

[5.2.3 搭建购物车列表条目界面布局 34](#_Toc91716511)

[5.2.4 搭建确认清空购物车界面布局 35](#_Toc91716512)

[5.2.5 编写店铺列表适配器 37](#_Toc91716513)

[5.2.6 编写店铺列表适配器 39](#_Toc91716514)

[5.2.7 编写店铺列表适配器 42](#_Toc91716515)

[5.3 菜品详情功能业务实现 50](#_Toc91716516)

[5.3.1 搭建菜品详情界面布局 50](#_Toc91716517)

[5.3.2 实现菜品界面显示功能 51](#_Toc91716518)

[5.4 订单功能业务实现 52](#_Toc91716519)

[5.4.1 搭建订单界面布局 53](#_Toc91716520)

[5.4.2 搭建订单列表条目界面布局 57](#_Toc91716521)

[5.4.3 搭建支付界面布局 59](#_Toc91716522)

[5.4.4 编写订单列表适配器 59](#_Toc91716523)

[5.4.5 实现订单显示与支付功能 62](#_Toc91716524)

[5.5 实现欢迎页面 64](#_Toc91716525)

[5.5.1 实现欢迎页面布局 64](#_Toc91716526)

[5.5.2 实现欢迎页面功能 65](#_Toc91716527)

[5.6 实现注册页面 67](#_Toc91716528)

[5.6.1 实现注册页面功能 67](#_Toc91716529)

[5.6.2 实现注册页面功能 69](#_Toc91716530)

[5.7 实现MD5加密算法存储账号密码 73](#_Toc91716531)

[5.8 实现登录页面 73](#_Toc91716532)

[5.8.1 实现登录页面功能 73](#_Toc91716533)

[5.8.2 实现登录页面功能 77](#_Toc91716534)

[6 遇到的问题与解决方法 80](#_Toc91716535)

[6.1 打开 AVD 时一直黑屏 80](#_Toc91716536)

[6.2 HAXM 安装错误 80](#_Toc91716537)

[6.3 测试类报错 Cannot resolve symbol 'Test' 80](#_Toc91716538)

[6.4 build 中文乱码问题 80](#_Toc91716539)

[6.5 okhttp3 3.14版本导致app闪退 80](#_Toc91716540)

[6.6 Tomcat 闪退 80](#_Toc91716541)

[6.7 报错 Resources$NotFoundException: String resource ID xxxx 81](#_Toc91716542)

[6.8 设置了 background 后，button 依旧是默认颜色 81](#_Toc91716543)

[6.9 华为、荣耀系列手机开始 ADB 调试后也不能连接到 Android Studio 81](#_Toc91716544)

[6.10 连接不到 tomcat，就连 OnFailure 函数都不进入 81](#_Toc91716545)

[7 参考文献 81](#_Toc91716546)

[8 实验感悟 81](#_Toc91716547)

# 课设目的与要求

## 课设目的

为了巩固所学Android基础知识，要开发一款仿美团外卖的项目，该项目与我们平常看到的美团外卖项目界面比较类似，展示的内容包括店铺、菜单、购物车、订单与支付等信息。

## 课设要求

1. 掌握Android应用程序开发中的基本技术。
2. 能分析仿美团外卖项目，能够说出项目的开发环境和模块。
3. 掌握服务器的搭建方式，能够独立搭建服务器。
4. 掌握店铺界面的开发过程，能够实现店铺界面的显示效果。
5. 掌握店铺详情界面与购物车的开发过程，能够实现购物车功能。
6. 掌握菜品详情界面的开发过程，能够实现菜品详情界面的功能。

# 课设概述

仿美团外卖项目是一个网上订餐项目，该项目中包含订餐的店铺、各店铺的菜单、购物车以及订单与付款等模块。在店铺列表中可以看到店铺的名称、月售数量、起送价格与配送费用、配送时间以及店铺特色等信息，点击店铺列表中的任意一个店铺，程序会进入到店铺详情界面，该界面主要用于显示店铺中的菜单信息，同时可以将想要吃的菜添加到购物车中，选完菜之后可以点击该界面中的“去结算”按钮，进入到订单界面，在该界面核对已点的菜单信息，并通过“去支付”按钮进行付款。

# 开发环境配置

## 操作系统

Windows 10 专业版 20H2

## 开发工具

Android Studio Arctic Fox | 2020.3.1 Patch 4

Tomcat 9.0

## API版本

API 选择 30 (R Android 11.0)

## 主要的库和依赖版本

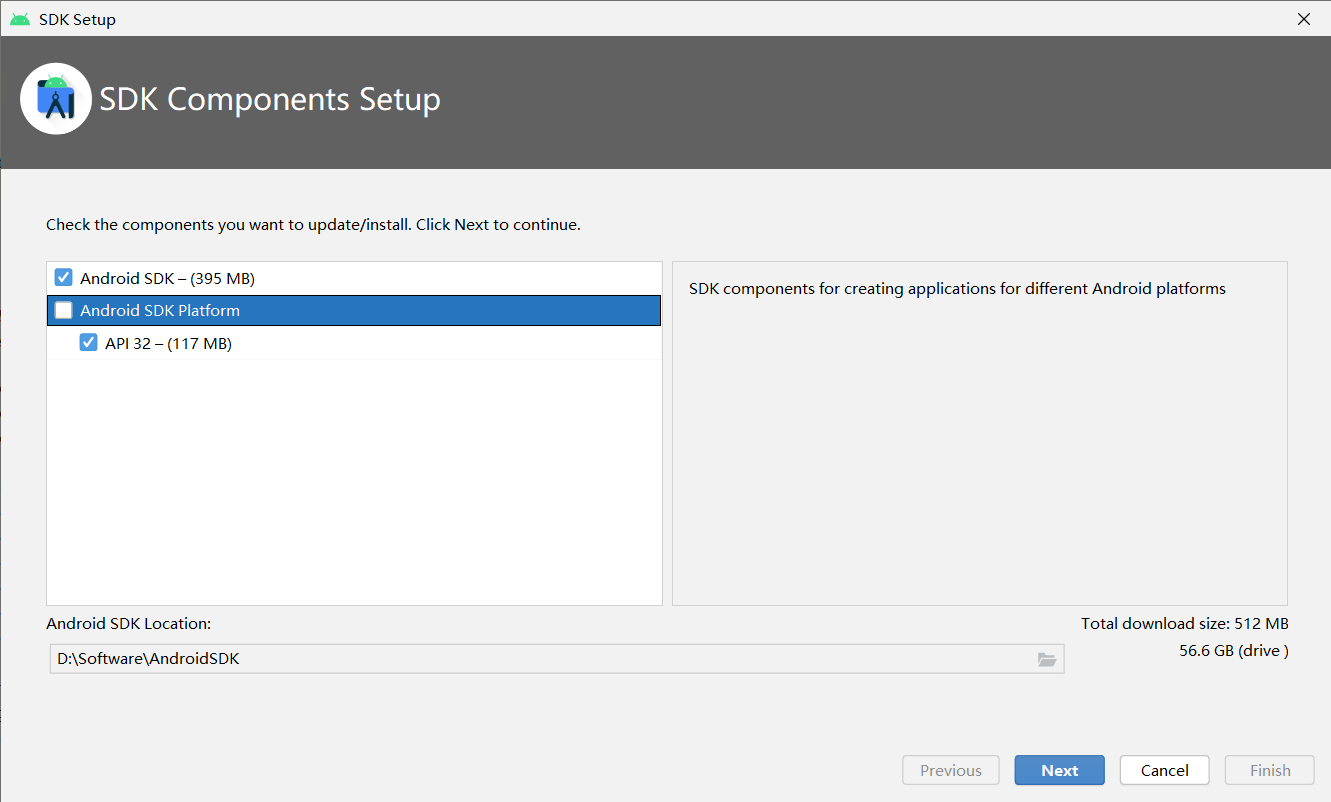
glide 3.7.0

gson 2.8.5

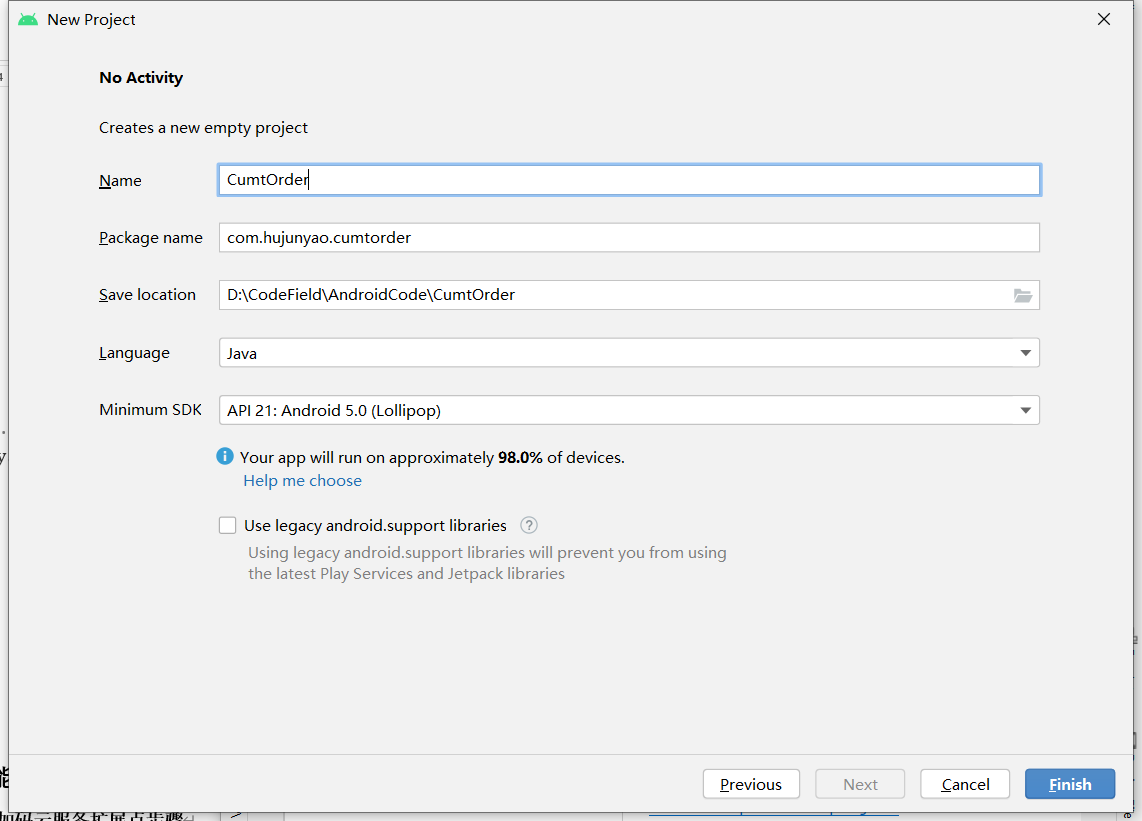
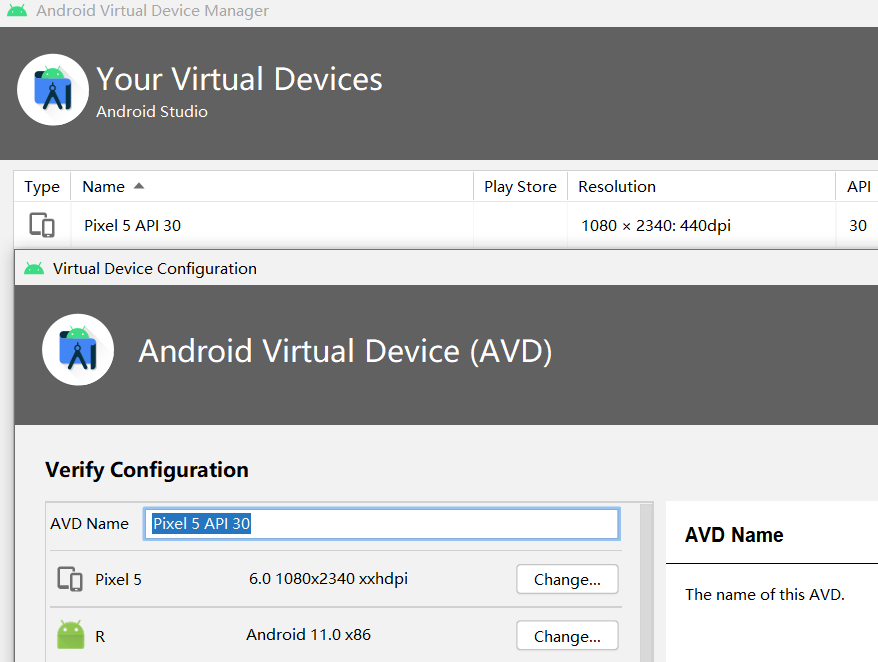
okhttp:5.0.0-alpha.3

# 所需环境与库安装准备过程

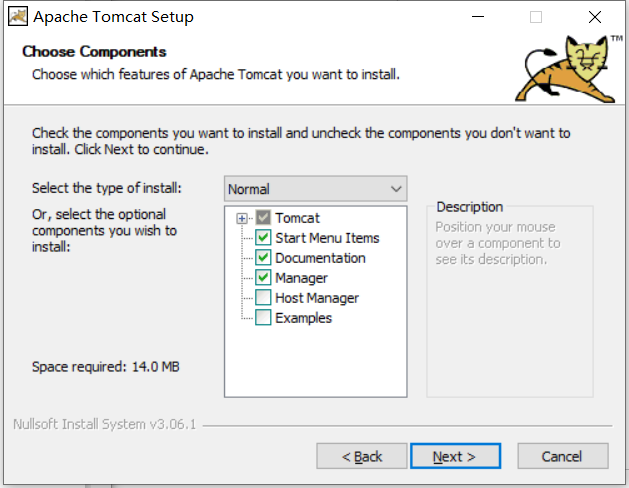
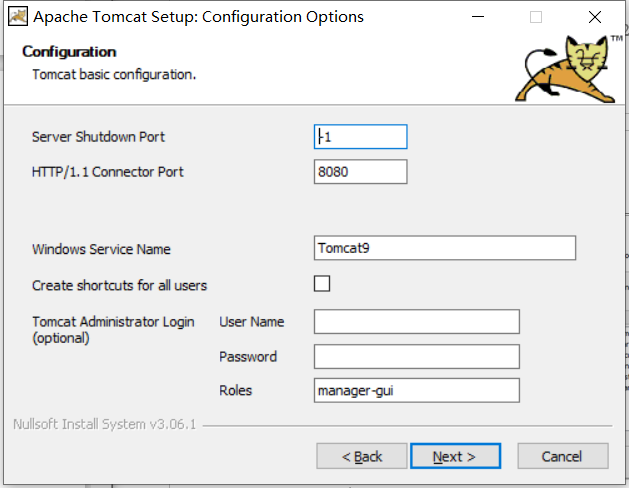
安装SDK，选择合适的目录。



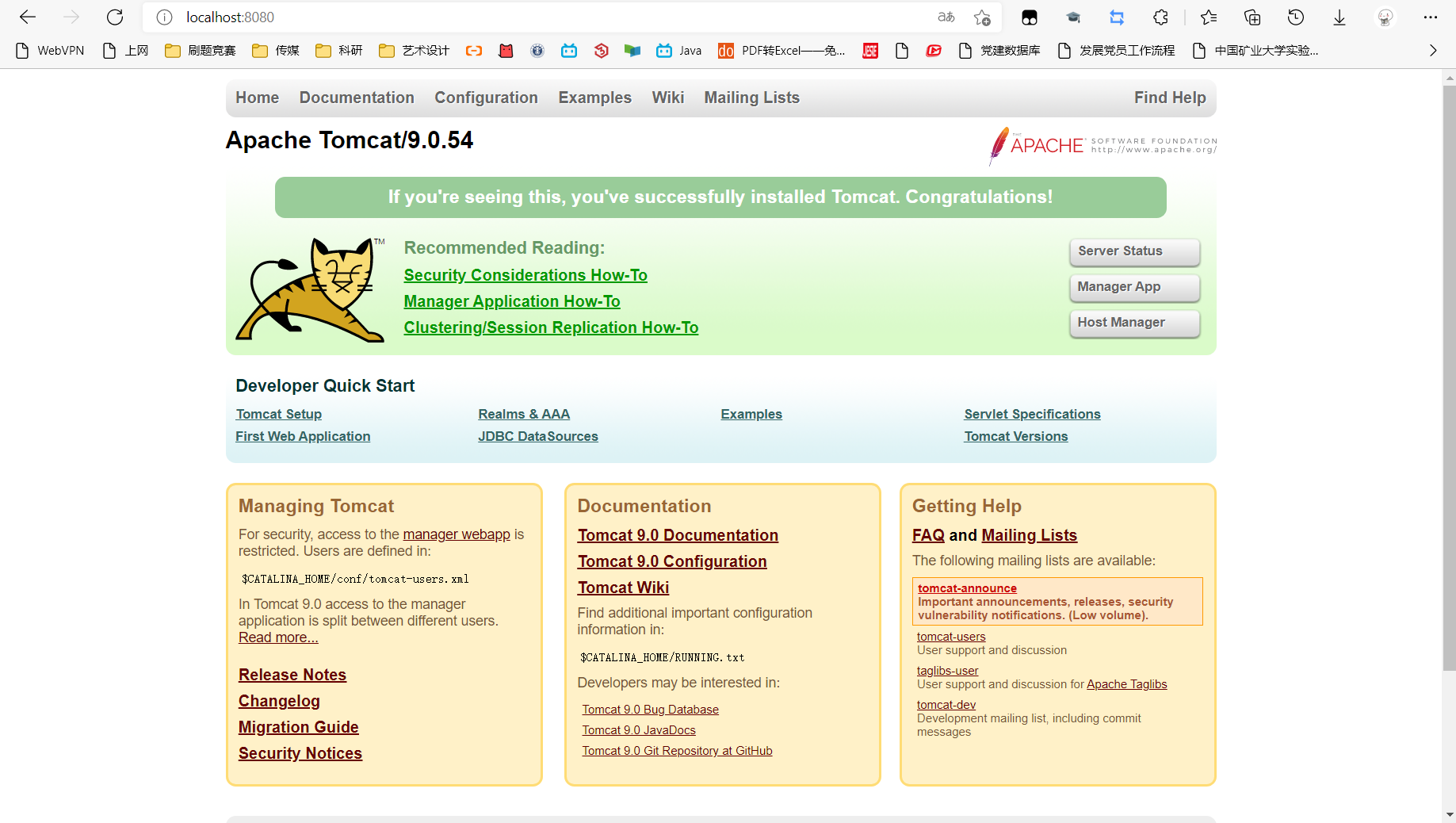
在 Android Studio 中选择【File -> New -> New Project】，选取 No Activity，创建名称为 CumtOrder 的Java工程，其他参数如下。点击可选择虚拟设备，选择 Pixel 5 作为虚拟机，API 选择 30.

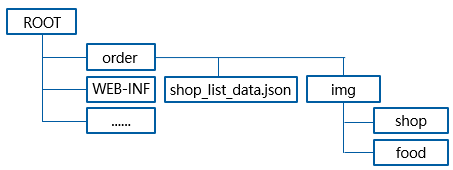
在 Tomcat 官网下载 Tomcat 9.0，进行安装操作，选择默认即可。

启动服务，在网址栏输入 localhost:8080，出现 Tomcat欢迎页面，经过验证，表明电脑可以成功连接到 Tomcat。



根据学习文档可知，ROOT 文件夹在 tomcat9\webapps 目录下，表示 Tomcat 的根目录。order 文件夹存放的是订餐项目用到的所有数据，其中， order/img 文件夹存放的是图片资源，包含店铺图片和菜单图片。shop\_list\_data.json 文件中存放的是店铺列表与店铺详情界面的数据，数据的内容摘录如下。



[

{

"id":1,

"shopName":"思维特精品小碗菜",

"saleNum":8544,

"offerPrice":10,

"distributionCost":0,

"welfare":"首单减6",

"time":50,

"shopPic":"/img/shop/shop1.png",

"shopNotice":"徐州小碗菜热销榜第一名",

"foodList":[

{

"foodId":1,

"foodName":"酸辣土豆丝",

"taste":"酸辣",

"saleNum":2601,

"price":1.98,

"count":0,

"foodPic":"/img/food/food11.png"

},

{

"foodId":2,

……

},

{

"foodId":3,

……

}

]

},

……

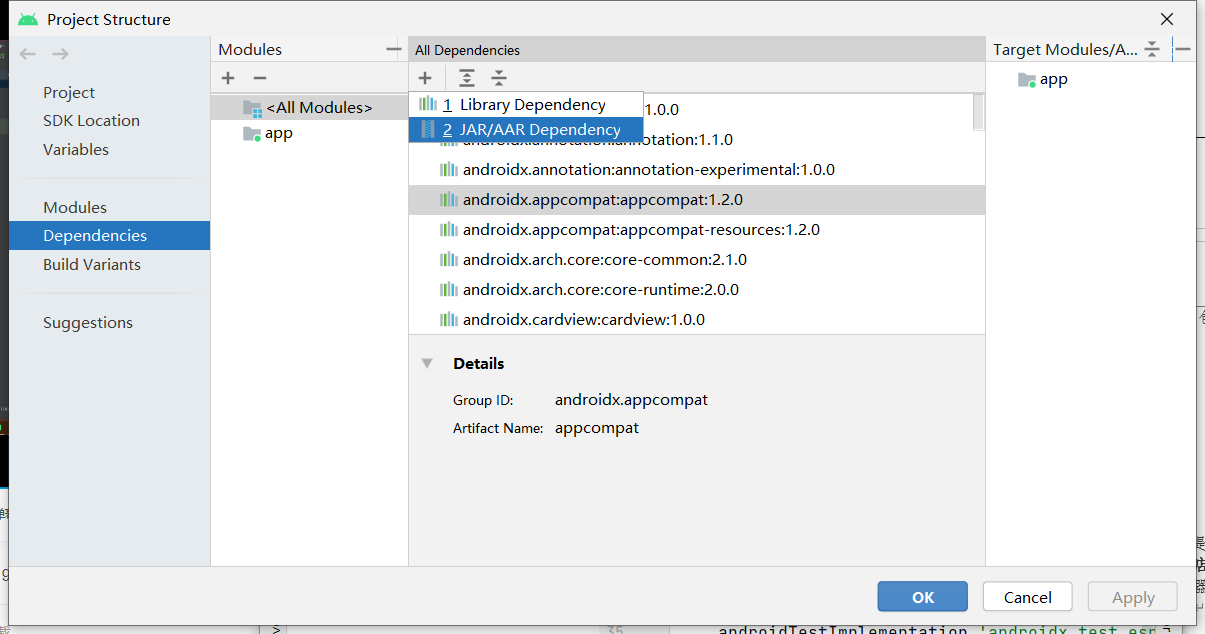
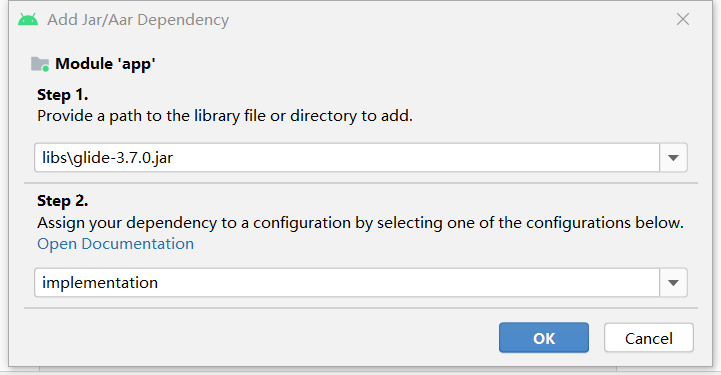
{

……

}

]]

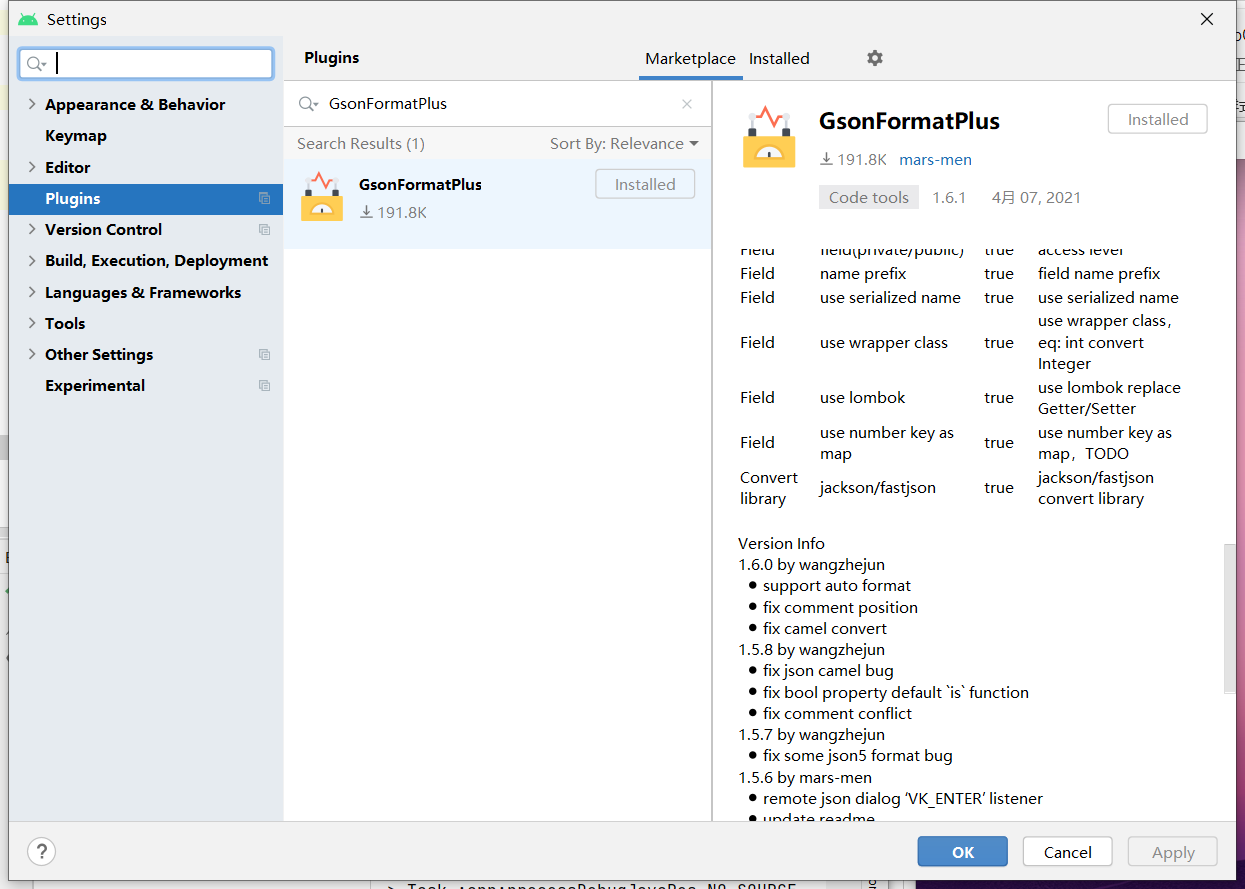
安装 jar 包的过程如下，以安装 glide 3.7.0.jar 为例，首先将包拖入 libs 文件夹，再打开【File -> Project Structure -> Dependencies】，点击加号，选择 JAR/AAR Dependency，选择需要的项目。

也可以直接链接网址导入。例如直接在 app 的 build.gradle 中添加 youth banner 2 的依赖 dependencies。最终项目的依赖状态如下。

dependencies **{** …  
 testImplementation 'junit:junit:4.13.2'  
 androidTestImplementation 'junit:junit:4.13.2'  
 implementation files('libs/glide-3.7.0.jar')  
 implementation files('libs/gson-2.8.5.jar')  
 implementation 'io.github.youth5201314:banner:2.2.2'  
 implementation 'com.squareup.okhttp3:okhttp:5.0.0-alpha.3'  
**}**

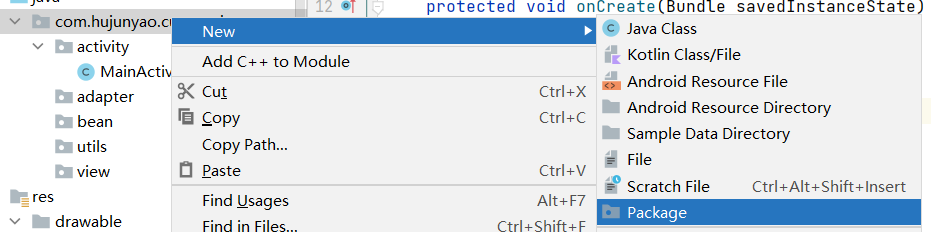
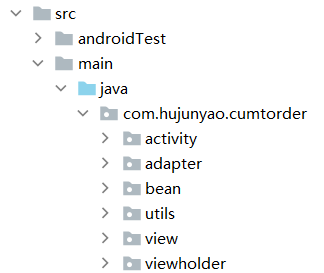
安装插件的过程如下，以安装 GsonFormatPlus 为例，打开【File -> Settings -> Plugins】，搜索 GsonFormatPlus，直接安装即可。



在【Project Structure -> Modules】中，可以直接设置编译的SDK的版本，构建工具版本等，也可以直接在 AndroidManifest.xml 修改。

android **{**  
 defaultConfig **{** applicationId "com.hujunyao.cumtorder"  
 minSdk 24  
 targetSdk 30  
 versionCode 1  
 versionName "1.0"  
 testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
 **}** compileOptions **{** sourceCompatibility 1.8  
 targetCompatibility 1.8  
 **}** compileSdkVersion 30  
 buildToolsVersion '30.0.2'  
**}**

为了使工程结构更加清晰，项目下设置不同的包来管理不同类型的类，如 activity, adapter, bean, utils, view, viewholder.

# 功能实现

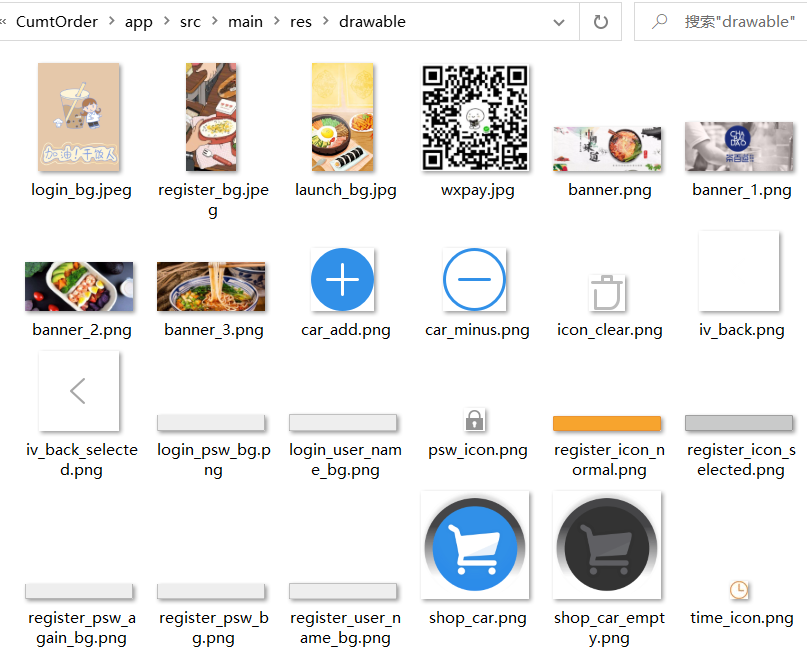
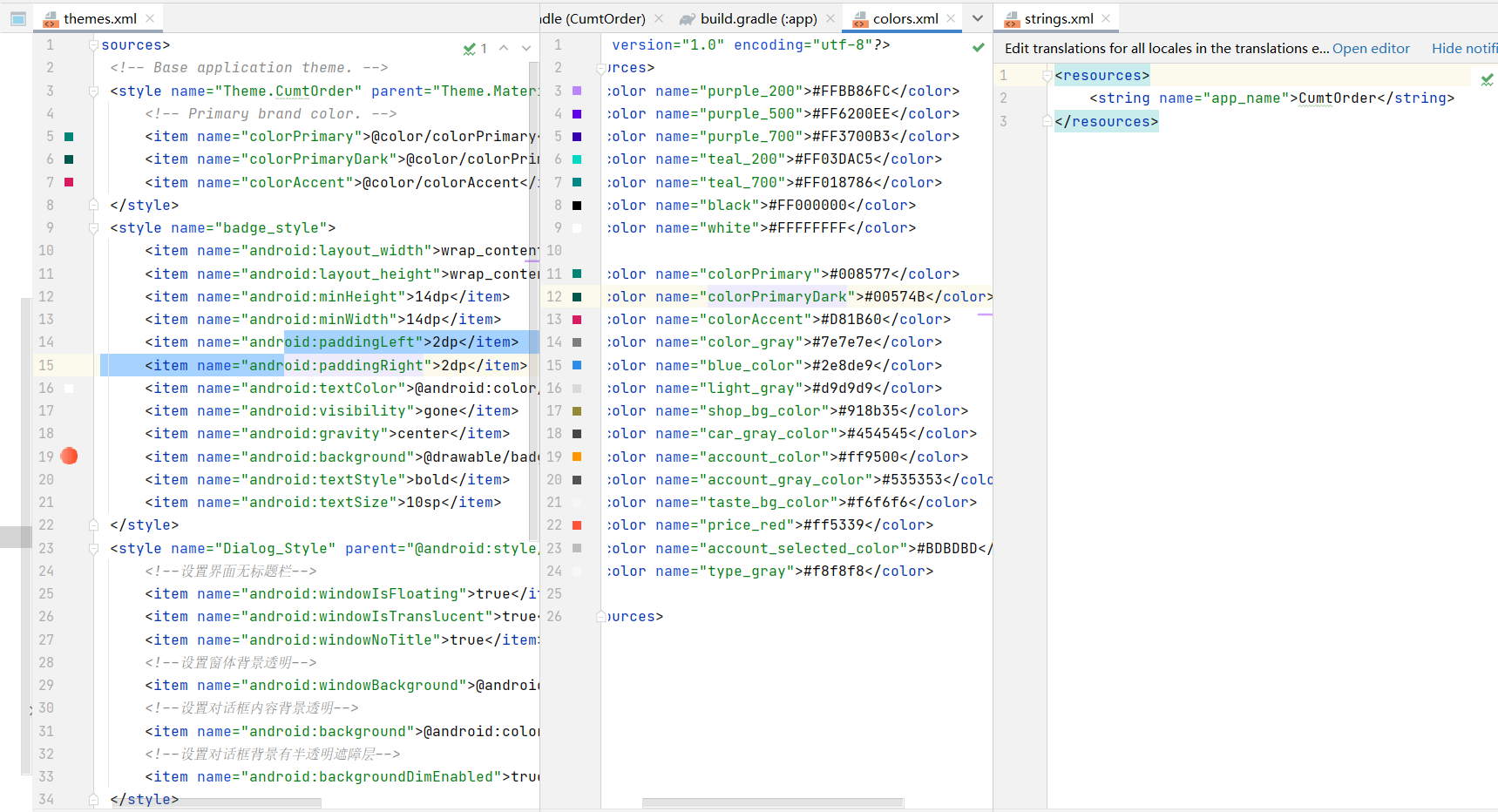
## 店铺功能业务实现

当打开仿美团外卖项目时，程序会直接进入主界面，也就是店铺列表界面。店铺列表界面从上至下分为**标题栏、水平滑动广告栏和店铺列表**三部分。其中，广告栏与店铺列表的数据是通过网络请求从服务器上获取的 Json 数据，接下来本节将针对店铺功能的相关业务进行开发。

### 搭建标题栏布局

创建项目：程序名 CumtOrder，包名为 com.hujunyao.cumtorder。（前面已完成该步骤）

提前进行各种导入：导入界面图片：将项目的 icon 图标 app\_icon.jpg 导入到 mipmap-hdpi 文件夹中。对于其他所需要的图片资源（返回、清空、加数量、减数量）加入 drawable 文件夹中。对于个人配色，将颜色导入 colors.xml 文件中。对于定制化样式，将样式导入 themes.xml 文件（以前是 styles.xml，注意可能版本不一致，名称不一样，但是功能是一样的，复制内容即可）中。导入动画，放入 anim 文件夹中。

放置界面控件，创建布局文件title\_bar.xml，放置2个TextView控件。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/title\_bar"  
 android:layout\_width="match\_parent" android:layout\_height="55dp"  
 android:background="@color/blue\_color">  
 <TextView  
 android:id="@+id/tv\_back"  
 android:layout\_width="55dp"  
 android:layout\_height="50dp"  
 android:layout\_centerVertical="true"  
 android:background="@drawable/go\_back\_selector"  
 android:clickable="true"  
 android:focusable="true"/>  
 <TextView  
 android:id="@+id/tv\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:textColor="@android:color/white"  
 android:textSize="25sp" />  
</RelativeLayout>

在 drawable 创建背景选择器 go\_back\_selector.xml。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<selector xmlns:android="http://schemas.android.com/apk/res/android">  
 <item android:drawable="@drawable/iv\_back" android:state\_pressed="false"/>  
 <item android:drawable="@drawable/iv\_back\_selected" android:state\_pressed="true"/>  
</selector>

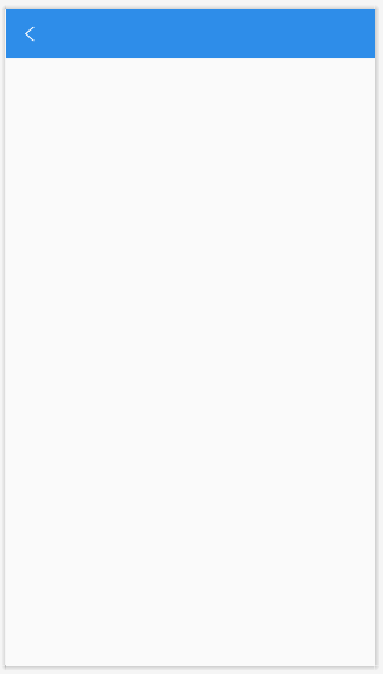
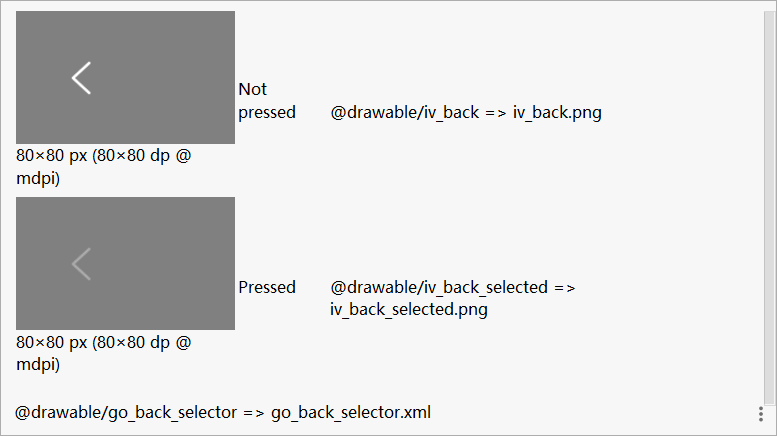
修改清单文件：在 AndroidManifest.xml 中引入程序的图标。

android:icon="@mipmap/app\_icon"  
android:roundIcon="@mipmap/app\_icon"

在 themes.xml 中去掉默认标题（一定要加 .Bridge）

<style name="Theme.CumtOrder" parent="Theme.MaterialComponents.DayNight.NoActionBar.Bridge">

当前效果如下。

### 搭建广告栏布局（轮播广告）

创建广告栏界面的布局文件 ad\_banner.xml。放置1个 ViewPager 控件，1个 ViewPagerIndicator 控件。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="160dp"  
 android:id="@+id/rl\_adBanner"  
 android:background="#eeeeee">  
 <androidx.viewpager.widget.ViewPager  
 android:id="@+id/vp\_advertBanner"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginBottom="1dp"  
 android:background="@drawable/banner"  
 android:gravity="center"/>  
 <LinearLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:background="@android:color/transparent">  
 <com.hujunyao.cumtorder.view.ViewPagerIndicator  
 android:id="@+id/vpi\_advert\_indicator"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:padding="4dp"/>  
 </LinearLayout>  
</RelativeLayout>

在 view 包中创建 ViewPagerIndicator 类并继承 LinearLayout 类。完整代码如下。

package com.hujunyao.cumtorder.view;  
import android.content.Context;  
import android.util.AttributeSet;  
import android.view.Gravity;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import com.hujunyao.cumtorder.R;  
public class ViewPagerIndicator extends LinearLayout {  
 private int mCount; *//小圆点的个数* private int mIndex; *//当前小圆点的位置* private Context context;  
 public ViewPagerIndicator(Context context) {  
 this(context,null);  
 }  
 public ViewPagerIndicator(Context context,AttributeSet attrs) {  
 super(context, attrs);  
 this.context = context;  
 setGravity(Gravity.*CENTER*); *//设置小圆点布局居中* }*//设置滑动到当前小圆点时其他小圆点的位置* public void setCurrentPosition(int currentIndex){  
 mIndex = currentIndex; *//当前小圆点* removeAllViews(); *//移除界面上存在的view* int pex = 5;  
 for (int i=0;i<mCount;i++){  
 ImageView imageView = new ImageView(context);  
 if (mIndex == i){ *//滑到当前界面  
 //蓝色小圆点* imageView.setImageResource(R.drawable.*indicator\_on*);  
 }else{  
 *//灰色小圆点* imageView.setImageResource(R.drawable.*indicator\_off*);  
 }  
 imageView.setPadding(pex,0,pex,0);  
 addView(imageView);  
 }  
 }*//设置小圆点的数目* public void setCount(int count){  
 this.mCount = count;  
 }  
}

创建 indicator\_on.xml 和 indicator\_off.xml。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="oval">  
 <size  
 android:width="6dp"  
 android:height="6dp" />  
 <solid android:color="#737373" />  
</shape>

*<?*xml version="1.0" encoding="utf-8"*?>*<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="oval">  
 <size  
 android:width="6dp"  
 android:height="6dp" />  
 <solid android:color="#00ABF8" />  
</shape>

### 搭建店铺页面布局

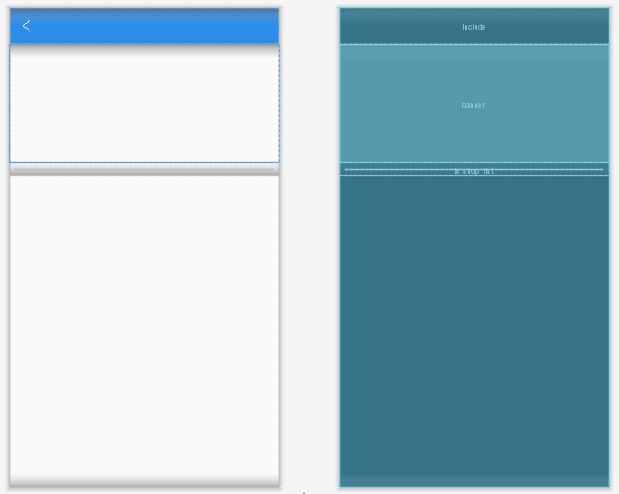
在 activity\_shop.xml 中引入 title\_bar.xml 文件，放置2个自定义控件 Banner 与 ShopListView。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".activity.ShopActivity"  
 android:orientation="vertical">  
 <include layout="@layout/title\_bar"/>  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <com.youth.banner.Banner  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/banner"  
 android:layout\_width="match\_parent"  
 android:layout\_height="180dp" />  
 <com.hujunyao.cumtorder.view.ShopListView  
 android:id="@+id/lv\_shop\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:divider="#FAFAFA"  
 android:dividerHeight="10dp"/>  
 </LinearLayout>  
 </ScrollView>  
</LinearLayout>

在 view 包创建 ShopListView 类继承 ListView 类。完整代码如下。

package com.hujunyao.cumtorder.view;  
  
import android.content.Context;  
import android.util.AttributeSet;  
import android.widget.ListView;  
  
public class ShopListView extends ListView {  
 public ShopListView(Context context) {  
 super(context);  
 }  
 public ShopListView(Context context, AttributeSet attrs) {  
 super(context, attrs);  
 }  
 public ShopListView(Context context, AttributeSet attrs, int defStyleAttr) {  
 super(context, attrs, defStyleAttr);  
 }  
 @Override  
 protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {  
 int heightSpec = MeasureSpec.*makeMeasureSpec*(Integer.*MAX\_VALUE* >> 2, MeasureSpec.*AT\_MOST*);  
 *//view 30位数，取Int最大值32位减少两位，模式是所有子控件高度* super.onMeasure(widthMeasureSpec, heightSpec);  
 }  
}

activity\_shop.xml 的效果如下。



### 搭建店铺列表条目界面布局

创建布局文件 shop\_item.xml。放置2个ImageView控件。放置6个TextView控件。

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="15dp"  
 android:background="@drawable/corner\_bg">  
  
 <ImageView  
 android:id="@+id/iv\_shop\_pic"  
 android:layout\_width="81dp"  
 android:layout\_height="81dp"  
 android:scaleType="fitXY" />  
  
 <LinearLayout  
 android:id="@+id/ll\_info"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_toEndOf="@id/iv\_shop\_pic"  
 android:layout\_marginStart="10dp"  
 android:orientation="vertical">  
 <TextView  
 android:id="@+id/tv\_shop\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="@android:color/black"  
 android:textSize="14sp" />  
 <TextView  
 android:id="@+id/tv\_sale\_num"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/color\_gray"  
 android:textSize="12sp" />  
 <TextView  
 android:id="@+id/tv\_cost"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/color\_gray"  
 android:textSize="12sp" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginTop="10dp">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@android:color/holo\_red\_light"  
 android:padding="2dp"  
 android:text="福利"  
 android:textColor="@color/white"  
 android:textSize="10sp" />  
  
 <TextView  
 android:id="@+id/tv\_welfare"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:textColor="@color/color\_gray"  
 android:textSize="10sp"/>  
 </LinearLayout>  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/tv\_time"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginTop="30dp"  
 android:textColor="@color/color\_gray"  
 android:layout\_centerVertical="true"  
 android:textSize="12sp"/>  
  
 <ImageView  
 android:layout\_width="15dp"  
 android:layout\_height="15dp"  
 android:layout\_marginEnd="10dp"  
 android:layout\_toStartOf="@+id/tv\_time"  
 android:background="@drawable/time\_icon"  
 android:layout\_centerVertical="true"  
 android:scaleType="fitXY"/>  
  
</RelativeLayout>

创建 corner\_bg.xml 文件，在该文件中设置一个边角为圆角的矩形。

*<?*xml version="1.0" encoding="utf-8"*?>*<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
 <solid android:color="@color/white" />  
 <corners android:radius="20dp" />  
 <stroke  
 android:width="1dp"  
 android:color="@color/light\_gray" />  
</shape>

创建背景选择器 item\_bg\_selector.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<selector xmlns:android="http://schemas.android.com/apk/res/android">  
 <item android:state\_pressed="true" >  
 <shape android:shape="rectangle">  
 <corners android:radius="10dp"/>  
 <solid android:color="#d4d4d4"/>  
 </shape>  
 </item>  
 <item android:state\_pressed="false" >  
 <shape android:shape="rectangle">  
 <corners android:radius="10dp"/>  
 <solid android:color="#ffffff" />  
 </shape>  
 </item>  
</selector>

### 封装店铺信息与菜品信息的实体类

由于店铺信息包含很多属性，因此，我们需要创建一个 ShopBean 类与封装店铺信息的属性。在 bean 包中创建一个 ShopBean 类。由于该类的对象中存储的信息需要在 Activity 之间进行传输，因此将ShopBean 类进行序列化，即实现 Serializable 接口。该类定义了店铺信息的所有属性。

package com.hujunyao.cumtorder.bean;  
  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.util.List;  
  
public class ShopBean implements Serializable {  
 private int id;  
 private String shopName;  
 private int saleNum;  
 private BigDecimal offerPrice;  
 private BigDecimal distributionCost;  
 private String welfare;  
 private int time;  
 private String shopPic;  
 private String shopNotice;  
 private List<FoodBean> foodList;  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getShopName() {  
 return shopName;  
 }  
  
 public void setShopName(String shopName) {  
 this.shopName = shopName;  
 }  
  
 public int getSaleNum() {  
 return saleNum;  
 }  
  
 public void setSaleNum(int saleNum) {  
 this.saleNum = saleNum;  
 }  
  
 public BigDecimal getOfferPrice() {  
 return offerPrice;  
 }  
  
 public void setOfferPrice(BigDecimal offerPrice) {  
 this.offerPrice = offerPrice;  
 }  
  
 public BigDecimal getDistributionCost() {  
 return distributionCost;  
 }  
  
 public void setDistributionCost(BigDecimal distributionCost) {  
 this.distributionCost = distributionCost;  
 }  
  
 public String getWelfare() {  
 return welfare;  
 }  
  
 public void setWelfare(String welfare) {  
 this.welfare = welfare;  
 }  
  
 public int getTime() {  
 return time;  
 }  
  
 public void setTime(int time) {  
 this.time = time;  
 }  
  
 public String getShopPic() {  
 return shopPic;  
 }  
  
 public void setShopPic(String shopPic) {  
 this.shopPic = shopPic;  
 }  
  
 public String getShopNotice() {  
 return shopNotice;  
 }  
  
 public void setShopNotice(String shopNotice) {  
 this.shopNotice = shopNotice;  
 }  
  
 public List<FoodBean> getFoodList() {  
 return foodList;  
 }  
  
 public void setFoodList(List<FoodBean> foodList) {  
 this.foodList = foodList;  
 }  
  
}

由于菜单列表包含很多属性，因此，我们需要创建一个 FoodBean 类封装菜单信息的属性。创建一个 FoodBean 类并实现 Serializable 接口，该类中定义了每个菜的所有属性。

package com.hujunyao.cumtorder.bean;  
import java.io.Serializable;  
import java.math.BigDecimal;  
public class FoodBean implements Serializable {  
 private int foodId;  
 private String foodName;  
 private String taste;  
 private int saleNum;  
 private BigDecimal price;  
 private int count;  
 private String foodPic;  
 public int getFoodId() {  
 return foodId;  
 }  
 public void setFoodId(int foodId) {  
 this.foodId = foodId;  
 }  
 public String getFoodName() {  
 return foodName;  
 }  
 public void setFoodName(String foodName) {  
 this.foodName = foodName;  
 }  
 public String getTaste() {  
 return taste;  
 }  
 public void setTaste(String taste) {  
 this.taste = taste;  
 }  
 public int getSaleNum() {  
 return saleNum;  
 }  
 public void setSaleNum(int saleNum) {  
 this.saleNum = saleNum;  
 }  
 public BigDecimal getPrice() {  
 return price;  
 }  
 public void setPrice(BigDecimal price) {  
 this.price = price;  
 }  
 public int getCount() {  
 return count;  
 }  
 public void setCount(int count) {  
 this.count = count;  
 }  
 public String getFoodPic() {  
 return foodPic;  
 }  
 public void setFoodPic(String foodPic) {  
 this.foodPic = foodPic;  
 }  
}

由于 Banner 列表包含很多属性，因此，我们需要创建一个 DataBean 类封装广告轮播信息的属性。创建一个 DataBean 类，该类中定义了每个广告的所有属性，用一个列表进行存储。

package com.hujunyao.cumtorder.bean;  
  
  
import com.hujunyao.cumtorder.R;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class DataBean {  
 public Integer imageRes;  
  
 public DataBean(Integer imageRes) {  
 this.imageRes = imageRes;  
 }  
  
 public static List<DataBean> getTestData() {  
 List<DataBean> list = new ArrayList<>();  
 list.add(new DataBean(R.drawable.*banner\_1*));  
 list.add(new DataBean(R.drawable.*banner\_2*));  
 list.add(new DataBean(R.drawable.*banner\_3*));  
 return list;  
 }  
  
}

### 编写广告栏的适配器

创建一个 adapter 包，并在该包中创建一个数据适配器 ImageAdapter 类， 继承 BannerAdapter。

package com.hujunyao.cumtorder.adapter;  
  
import android.view.ViewGroup;  
import android.widget.ImageView;  
  
import com.hujunyao.cumtorder.bean.DataBean;  
import com.hujunyao.cumtorder.viewholder.ImageHolder;  
import com.youth.banner.adapter.BannerAdapter;  
  
import java.util.List;  
  
*/\*\*  
 \* 自定义布局，图片  
 \*/*public class ImageAdapter extends BannerAdapter<DataBean, ImageHolder> {  
  
 public ImageAdapter(List<DataBean> mDatas) {  
 *//设置数据，也可以调用banner提供的方法,或者自己在adapter中实现* super(mDatas);  
 }  
 *//创建ViewHolder，可以用viewType这个字段来区分不同的ViewHolder* @Override  
 public ImageHolder onCreateHolder(ViewGroup parent, int viewType) {  
 ImageView imageView = new ImageView(parent.getContext());  
 *//注意，必须设置为match\_parent，这个是viewpager2强制要求的* ViewGroup.LayoutParams params = new ViewGroup.LayoutParams(  
 ViewGroup.LayoutParams.*MATCH\_PARENT*,  
 ViewGroup.LayoutParams.*MATCH\_PARENT*);  
 imageView.setLayoutParams(params);  
 imageView.setScaleType(ImageView.ScaleType.*CENTER\_CROP*);  
 return new ImageHolder(imageView);  
 }  
  
 @Override  
 public void onBindView(ImageHolder holder, DataBean data, int position, int size) {  
 holder.imageView.setImageResource(data.imageRes);  
 }  
  
}

### 编写店铺列表适配器

由于店铺界面的列表是用 ShopListView 控件展示的，所以需要创建一个数据适配器 ShopAdapter 对 ShopListView 控件进行数据适配。

创建一个店铺列表的适配器 ShopAdapter，在该适配器中重写getCount() 方法、 getItem() 方法、 getItemId() 方法和 getView() 方法，这些方法分别用于获取列表中条目的总数、对应的条目对象、条目对象的 Id 、对应的条目视图。为了减少程序的缓存，需要在 getView() 方法中复用 convertView 对象。

package com.hujunyao.cumtorder.adapter;  
  
import static com.hujunyao.cumtorder.utils.Constant.*WEB\_SITE*;  
  
import android.content.Context;  
import android.content.Intent;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.bumptech.glide.Glide;  
import com.hujunyao.cumtorder.R;  
import com.hujunyao.cumtorder.activity.ShopDetailActivity;  
import com.hujunyao.cumtorder.bean.ShopBean;  
  
import java.util.List;  
  
public class ShopAdapter extends BaseAdapter {  
 private final Context context;  
 private List<ShopBean> sbl;  
  
 public ShopAdapter(Context context) { this.context = context; }  
  
 public void setData(List<ShopBean> sbl) {  
 this.sbl = sbl;  
 notifyDataSetChanged();  
 }  
  
 @Override  
 public int getCount() { return sbl == null ? 0 : sbl.size(); }  
  
 @Override  
 public ShopBean getItem(int position) { return sbl == null ? null : sbl.get(position);}  
  
 @Override  
 public long getItemId(int position) { return position; }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 ViewHolder vh = null;  
 *//复用convertView* if (convertView == null) {  
 convertView = LayoutInflater.*from*(context).inflate(R.layout.*shop\_item*,null);  
 vh = new ViewHolder();  
 vh.tv\_shop\_name = (TextView) convertView.findViewById(R.id.*tv\_shop\_name*);  
 vh.tv\_sale\_num = (TextView) convertView.findViewById(R.id.*tv\_sale\_num*);  
 vh.tv\_cost = (TextView) convertView.findViewById(R.id.*tv\_cost*);  
 vh.tv\_welfare = (TextView) convertView.findViewById(R.id.*tv\_welfare*);  
 vh.tv\_time = (TextView) convertView.findViewById(R.id.*tv\_time*);  
 vh.iv\_shop\_pic = (ImageView) convertView.findViewById(R.id.*iv\_shop\_pic*);  
 convertView.setTag(vh);  
 } else {  
 vh = (ViewHolder) convertView.getTag();  
 }  
 *//获取position对应的Item的数据对象* ShopBean bean = getItem(position);  
 if (bean != null) {  
 vh.tv\_shop\_name.setText(bean.getShopName());  
 vh.tv\_sale\_num.setText("月售" + bean.getSaleNum());  
 vh.tv\_cost.setText("起送￥" + bean.getOfferPrice() + "|配送￥" +  
 bean.getDistributionCost());  
 vh.tv\_time.setText("大约用时"+bean.getTime()+"分钟");  
 vh.tv\_welfare.setText(bean.getWelfare());  
 Glide.*with*(context)  
 .load(*WEB\_SITE* + bean.getShopPic())  
 .error(R.mipmap.*ic\_launcher*)  
 .into(vh.iv\_shop\_pic);  
 }  
 *//每个Item的点击事件* convertView.setOnClickListener(v -> {  
 *//跳转到店铺详情界面* if (bean == null) return;  
 Intent intent = new Intent(context, ShopDetailActivity.class);  
 *//把店铺的详细信息传递到店铺详情界面* intent.putExtra("shop", bean);  
 context.startActivity(intent);  
  
 });  
 return convertView;  
 }  
 static class ViewHolder {  
 TextView tv\_shop\_name, tv\_sale\_num, tv\_cost, tv\_welfare, tv\_time;  
 ImageView iv\_shop\_pic;  
 }  
}

### 实现店铺界面显示功能

添加 okhttp 库、添加 gson 库（已经实现）。

创建Constant类由于仿美团外卖项目中的数据需要通过请求网络从Tomcat（一个小型服务器）上获取，所以需要创建一个Constant类存放各界面从服务器上请求数据时使用的接口地址。完整代码如下。（IP根据自己的电脑填写，可以在 cmd 中输入 ipconfig 获取 IP 地址）

package com.hujunyao.cumtorder.utils;  
public class Constant {  
 public static final String *WEB\_SITE* = "http://10.3.xxx.xxx:8080/order";*//内网接口* public static final String *REQUEST\_SHOP\_URL* = "/shop\_list\_data.json"; *//店铺列表接口*}

创建 JsonParse 类，由于从 Tomcat 服务器上获取的店铺数据是JSON 类型的数据，JSON 数据不能直接显示到界面上，所以需要在包中创建一个 JsonParse 类用于解析获取到的 JSON 数据。

package com.hujunyao.cumtorder.utils;  
  
import com.google.gson.Gson;  
import com.google.gson.reflect.TypeToken;  
import com.hujunyao.cumtorder.bean.ShopBean;  
  
import java.lang.reflect.Type;  
import java.util.List;  
  
public class JsonParse {  
 private static JsonParse *instance*;  
 *//构造方法私有化，让别人没办法new* private JsonParse() {}  
 *//单例模式* public static JsonParse getInstance() {  
 if (*instance* == null) {  
 *instance* = new JsonParse();  
 }  
 return *instance*;  
 }  
 public List<ShopBean> getShopList(String json) {  
 *// 创建一个TypeToken的匿名子类对象，并调用对象的getType()方法* Type listType = new TypeToken<List<ShopBean>>() {}.getType();  
 Gson gson = new Gson(); *// 使用gson库解析JSON数据  
 // 把获取到的信息集合存到shopList中* List<ShopBean> shopList = gson.fromJson(json, listType);  
 return shopList;  
 }  
}

将数据显示到店铺界面上。修改 ShopActivity 类。

private void initView() {  
 tvBack = (TextView) findViewById(R.id.*tv\_back*);  
 tvTitle = findViewById(R.id.*tv\_title*);  
 lvShopList = findViewById(R.id.*lv\_shop\_list*);  
 adapter = new ShopAdapter(this);  
 lvShopList.setAdapter(adapter);  
 tvTitle.setText("首页");  
 tvBack.setVisibility(View.*GONE*);  
  
 banner = (Banner) findViewById(R.id.*banner*);  
  
 ImageAdapter bannerImageAdapter = new ImageAdapter(DataBean.*getTestData*());  
  
 banner.setAdapter(bannerImageAdapter)*//设置适配器* .setCurrentItem(0,false)  
 .addBannerLifecycleObserver(this)*//添加生命周期观察者* .addPageTransformer(new ScaleInTransformer())*//添加切换效果* .setIndicator(new CircleIndicator(this));*//设置指示器* }  
  
 private void initData() {  
 OkHttpClient okHttpClient = new OkHttpClient();  
 Request request = new Request.Builder().url(Constant.*WEB\_SITE* + Constant.*REQUEST\_SHOP\_URL*).build();  
*// Request request = new Request.Builder().url("http://10.0.2.2:8080/order/shop\_list\_data.json").build();* Call call = okHttpClient.newCall(request);  
 *// 异步线程访问网络* call.enqueue(new Callback() {  
 @Override  
 public void onResponse(Call call, Response response) throws IOException {  
 String json = Objects.*requireNonNull*(response.body()).string(); *//获取店铺数据* Log.*v*("tag", "json: " + json);  
 final List<ShopBean> list = JsonParse.*getInstance*().getShopList(json);  
 runOnUiThread(() -> adapter.setData(list));  
 }  
 @Override  
 public void onFailure(Call call, IOException e) {  
 Log.*v*("tag", "你这网络不行啊" );  
 }  
 });  
 }  
  
 @Override  
 public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) { }  
  
 @Override  
 public void onPageSelected(int position) { }  
  
 @Override  
 public void onPageScrollStateChanged(int state) { }

## 店铺功能业务实现

### 实现店铺界面显示功能

创建 ShopDetailActivity，指定布局文件名为 activity\_shop\_detail，放置1个 TextView 控件、1个 View 控件、1个 ListView 控件。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<FrameLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <LinearLayout  
 android:id="@+id/ll\_intro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/blue\_color"  
 android:orientation="vertical">  
 <include layout = "@layout/title\_bar"/>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="60dp"  
 android:background="@android:color/white"  
 android:orientation="vertical"  
 android:paddingTop="60dp">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:text="菜单"  
 android:textColor="@android:color/black"  
 android:textSize="16sp" />  
 <View  
 android:layout\_width="match\_parent"  
 android:layout\_height="1dp"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginEnd="10dp"  
 android:background="@color/light\_gray" />  
 </LinearLayout>  
 </LinearLayout>  
 <include layout="@layout/shop\_detail\_head"/>  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="220dp">  
 <ListView  
 android:id="@+id/lv\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginBottom="50dp"  
 android:layout\_marginTop="10dp" />  
 <include layout="@layout/car\_list" />  
 <include layout="@layout/shop\_car" />  
 </RelativeLayout>  
</FrameLayout>

创建布局文件 shop\_detail\_head.xml。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/type\_gray"  
 android:orientation="vertical">  
 <include layout="@layout/title\_bar" />  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="15dp"  
 android:background="@android:color/white">  
  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:text="收货地址"  
 android:textColor="@color/color\_gray"  
 android:textSize="16sp"  
 android:layout\_weight="2"/>  
  
 <EditText  
 android:id="@+id/et\_address"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:background="@android:color/white"  
 android:hint="如：矿大南湖校区计算机学院"  
 android:scrollbars="horizontal"  
 android:singleLine="true"  
 android:textColor="@android:color/black"  
 android:textSize="16sp"  
 android:layout\_weight="7"/>  
 <ImageView  
 android:id="@+id/iv\_address"  
 android:layout\_width="0dp"  
 android:layout\_height="20dp"  
 android:src="@drawable/car\_add"  
 android:layout\_gravity="center\_vertical"  
 android:layout\_weight="1"  
 />  
 </LinearLayout>  
  
 <ListView  
 android:id="@+id/lv\_order"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:background="@android:color/white" >  
 </ListView>  
 <View  
 android:layout\_width="match\_parent"  
 android:layout\_height="1dp"  
 android:background="@color/type\_gray" />  
 <RelativeLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@android:color/white">  
 <TextView  
 android:id="@+id/tv\_cost"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:padding="10dp"  
 android:textColor="@color/price\_red"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="2dp"  
 android:layout\_toStartOf="@id/tv\_cost"  
 android:padding="10dp"  
 android:text="小计"  
 android:textColor="@android:color/black"  
 android:textSize="14sp" />  
 <TextView  
 android:id="@+id/tv\_distribution\_cost"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_below="@id/tv\_cost"  
 android:padding="10dp"  
 android:textColor="@android:color/black"  
 android:textSize="14sp"  
 android:textStyle="bold" />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/tv\_cost"  
 android:layout\_toStartOf="@id/tv\_distribution\_cost"  
 android:padding="10dp"  
 android:text="配送费"  
 android:textColor="@android:color/black"  
 android:textSize="14sp" />  
 </RelativeLayout>  
</LinearLayout>

创建布局文件 shop\_car.xml。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="65dp"  
 android:layout\_alignParentBottom="true">  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="15dp"  
 android:background="@color/car\_gray\_color"  
 android:paddingStart="60dp">  
 <TextView  
 android:id="@+id/tv\_money"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="25dp"  
 android:layout\_marginStart="4dp"  
 android:layout\_marginTop="4dp"  
 android:gravity="center"  
 android:text="未选购商品"  
 android:textColor="@color/light\_gray"  
 android:textSize="16sp" />  
 <TextView  
 android:id="@+id/tv\_distribution\_cost"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/tv\_money"  
 android:layout\_marginStart="4dp"  
 android:textColor="@android:color/white"  
 android:textSize="12sp"  
 android:textStyle="bold"  
 android:visibility="gone" />  
 <TextView  
 android:id="@+id/tv\_settle\_accounts"  
 android:layout\_width="100dp"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentEnd="true"  
 android:layout\_centerVertical="true"  
 android:background="@color/account\_color"  
 android:gravity="center"  
 android:text="去结算"  
 android:textColor="@android:color/white"  
 android:textSize="16sp"  
 android:visibility="gone" />  
  
 <TextView  
 android:id="@+id/tv\_not\_enough"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentEnd="true"  
 android:background="@color/account\_gray\_color"  
 android:gravity="center"  
 android:padding="8dp"  
 android:textColor="@color/light\_gray"  
 android:textSize="16sp" />  
 </RelativeLayout>  
 <include layout="@layout/car" />  
 </FrameLayout>  
</RelativeLayout>

创建布局文件 car\_list.xml。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/rl\_car\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginBottom="65dp"  
 android:gravity="bottom"  
 android:orientation="vertical"  
 android:visibility="gone"  
 >  
 <LinearLayout  
 android:id="@+id/ll\_intro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="32dp"  
 android:background="@color/light\_gray"  
 android:gravity="center\_vertical"  
 android:orientation="horizontal"  
 android:paddingLeft="8dp"  
 android:paddingRight="8dp">  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:layout\_weight="1"  
 android:text="已选商品"  
 android:textColor="@color/color\_gray"  
 android:textSize="12sp"  
 android:textStyle="bold" />  
 <TextView  
 android:id="@+id/tv\_clear"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:drawablePadding="2dp"  
 android:gravity="center"  
 android:text="清空"  
 android:textColor="@color/color\_gray"  
 android:textSize="12sp" />  
 <ImageView  
 android:id="@+id/iv\_clear"  
 android:layout\_width="14dp"  
 android:layout\_height="14dp"  
 android:background="@drawable/icon\_clear" />  
 </LinearLayout>  
 <ListView  
 android:id="@+id/lv\_car"  
 android:layout\_below="@id/ll\_intro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
</RelativeLayout>

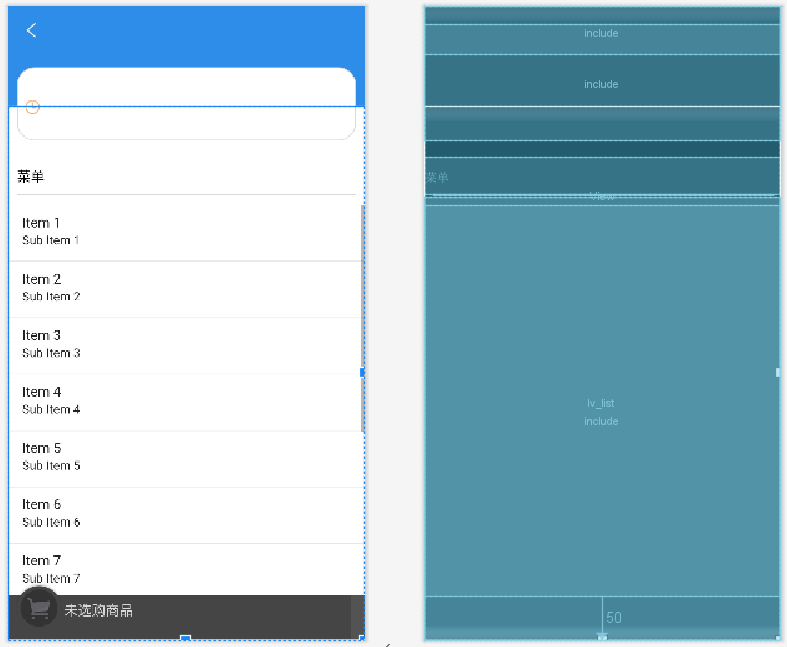
创建 corner\_bg.xml 文件。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
 <solid android:color="@color/white" />  
 <corners android:radius="20dp" />  
 <stroke  
 android:width="1dp"  
 android:color="@color/light\_gray" />  
</shape>

创建 badge\_bg.xml 文件。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
 <gradient  
 android:endColor="#fe451d"  
 android:startColor="#fe957f"  
 android:type="linear" />  
 <corners android:radius="180dp" />  
</shape>

activity\_shop\_detail.xml 界面如下。



### 搭建菜单列表条目界面布局

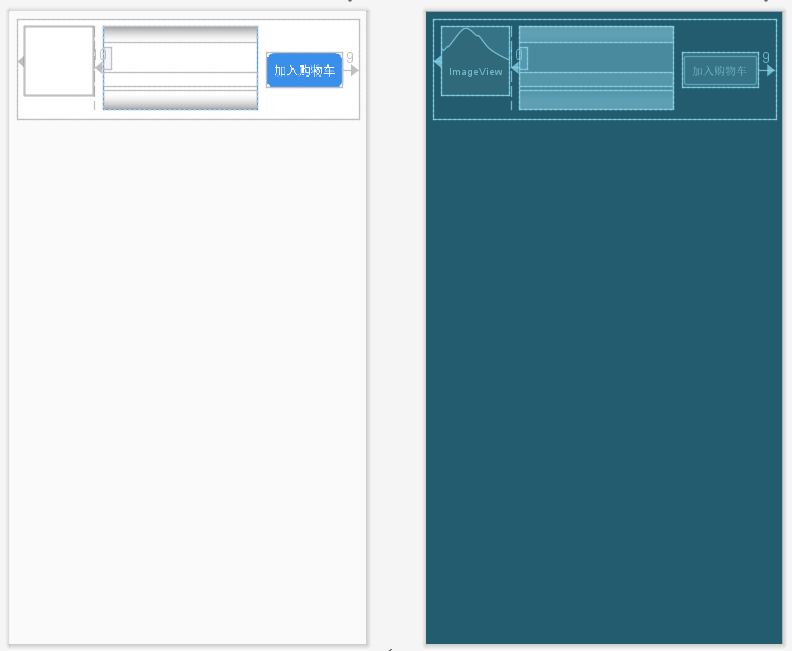
创建布局文件 menu\_item.xml。放置4个TextView控件、1个ImageView控件、1个Button控件。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginTop="8dp"  
 android:background="@drawable/menu\_item\_bg\_selector"  
 android:padding="10dp">  
 <ImageView  
 android:id="@+id/iv\_food\_pic"  
 android:layout\_width="80dp"  
 android:layout\_height="80dp"  
 android:layout\_alignParentStart="true"  
 android:scaleType="fitXY" />  
 <LinearLayout  
 android:layout\_width="178dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_toEndOf="@id/iv\_food\_pic"  
 android:orientation="vertical">  
 <TextView  
 android:id="@+id/tv\_food\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="@android:color/black"  
 android:textSize="14sp" />  
 <TextView  
 android:id="@+id/tv\_taste"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="4dp"  
 android:background="@drawable/taste\_bg"  
 android:padding="4dp"  
 android:textColor="@android:color/black"  
 android:textSize="12sp" />  
 <TextView  
 android:id="@+id/tv\_sale\_num"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="4dp"  
 android:textColor="@color/color\_gray"  
 android:textSize="12sp" />  
 <TextView  
 android:id="@+id/tv\_price"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="4dp"  
 android:textColor="@color/price\_red"  
 android:textSize="16sp" />  
 </LinearLayout>  
 <Button  
 android:id="@+id/btn\_add\_car"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="40dp"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginEnd="9dp"  
 android:background="@drawable/add\_car\_selector"  
 android:gravity="center"  
 android:text="加入购物车"  
 android:textColor="@android:color/white"  
 android:textSize="14sp" />  
</RelativeLayout>

创建背景选择器 menu\_item\_bg\_selector.xml。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<selector xmlns:android="http://schemas.android.com/apk/res/android">  
 <item android:drawable="@color/light\_gray" android:state\_pressed="true"/>  
  
 <item android:drawable="@android:color/white" android:state\_pressed="false"/>  
</selector>

menu\_item.xml 界面如下。

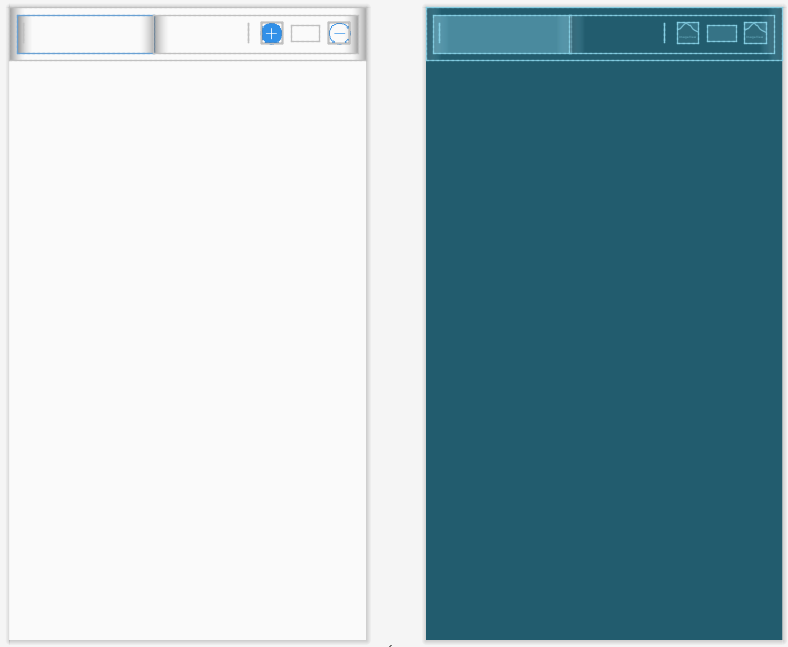


### 搭建购物车列表条目界面布局

创建布局文件 car\_item.xml。放置3个TextView控件、2个ImageView控件。在res文件夹中创建anim文件夹，在该文件夹中创建slide\_bottom\_to\_top.xml文件（已创建）完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="60dp"  
 android:padding="8dp">  
  
 <LinearLayout  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:gravity="center\_vertical"  
 android:layout\_weight="2">  
  
 <TextView  
 android:id="@+id/tv\_food\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:maxLines="1"  
 android:textColor="#757575"  
 android:textSize="16sp" />  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:gravity="right|center\_vertical"  
 android:layout\_weight="3">  
  
 <TextView  
 android:id="@+id/tv\_food\_price"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="15dp"  
 android:textColor="@color/price\_red"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
 <ImageView  
 android:id="@+id/iv\_add"  
 android:layout\_width="25dp"  
 android:layout\_height="25dp"  
 android:src="@drawable/car\_add" />  
  
 <TextView  
 android:id="@+id/tv\_food\_count"  
 android:layout\_width="34dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginEnd="10dp"  
 android:gravity="center"  
 android:textColor="@android:color/black" />  
  
 <ImageView  
 android:id="@+id/iv\_minus"  
 android:layout\_width="25dp"  
 android:layout\_height="25dp"  
 android:layout\_marginEnd="10dp"  
 android:src="@drawable/car\_minus" />  
  
 </LinearLayout>  
</LinearLayout>

car\_item.xml 界面如下。



### 搭建确认清空购物车界面布局

创建布局文件 dialog\_clear.xml。放置3个 TextView 控件。在styles.xml文件中添加一个名为 Dialog\_Style 的样式。（已导入）完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="250dp"  
 android:layout\_height="100dp"  
 android:background="@android:color/white"  
 android:gravity="center"  
 android:orientation="vertical">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_horizontal"  
 android:text="确认清空购物车？"  
 android:textColor="@color/color\_gray"  
 android:textSize="16sp" />  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="15dp"  
 android:gravity="center\_horizontal"  
 android:orientation="horizontal">  
 <TextView  
 android:id="@+id/tv\_cancel"  
 android:layout\_width="80dp"  
 android:layout\_height="30dp"  
 android:background="@drawable/item\_bg\_selector"  
 android:gravity="center"  
 android:text="取消"  
 android:textColor="@color/blue\_color"  
 android:textSize="14sp" />  
 <TextView  
 android:id="@+id/tv\_clear"  
 android:layout\_width="80dp"  
 android:layout\_height="30dp"  
 android:layout\_marginLeft="20dp"  
 android:background="@drawable/item\_bg\_selector"  
 android:gravity="center"  
 android:text="清空"  
 android:textColor="@color/blue\_color"  
 android:textSize="14sp" />  
 </LinearLayout>  
</LinearLayout>

dialog\_clear.xml 界面如下。



### 编写店铺列表适配器

由于店铺详情界面中的菜单列表是用 ListView 控件展示的，所以需要创建一个数据适配器 MenuAdapter 对 ListView 控件进行数据适配。编写菜单列表适配器的具体步骤如下。

创建菜单列表适配器 MenuAdapter。在该包中创建一个菜单列表适配器 MenuAdapter，并在该适配器中重写 getCount() 方法、getItem() 方法、getItemId() 方法和 getView() 方法。

创建 ViewHolder 类。MenuAdapter 中创建 ViewHolder 类，该类主要用于定义菜单列表条目上的控件对象，当菜单列表快速滑动时，该类可以快速为界面控件设置值，而不必每次重新创建很多控件对象，从而有效提高程序的性能。

.创建 OnSelectListener 接口。当点击菜单列表上的“加入购物车”按钮时，会增加购物车中菜品的数量，该数量的增加需要在 ShopDetailActivity 中进行，所以需要在 MenuAdapter 中创建一个 OnSelectListener 接口，在该接口中创建一个 onSelectAddCar() 方法用于处理“加入购物车”按钮的点击事件。

完整代码如下。

package com.hujunyao.cumtorder.adapter;  
  
import static com.hujunyao.cumtorder.utils.Constant.*WEB\_SITE*;  
  
import android.content.Context;  
import android.content.Intent;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.bumptech.glide.Glide;  
import com.hujunyao.cumtorder.R;  
import com.hujunyao.cumtorder.activity.FoodActivity;  
import com.hujunyao.cumtorder.bean.FoodBean;  
  
import java.util.List;  
  
  
public class MenuAdapter extends BaseAdapter {  
 private final Context context;  
 private List<FoodBean> fbl; *//菜单列表数据* private OnSelectListener onSelectListener; *//加入购物车按钮的监听事件* public MenuAdapter(Context context, OnSelectListener onSelectListener) {  
 this.context = context;  
 this.onSelectListener=onSelectListener;  
 }  
  
 public void setData(List<FoodBean> fbl) {  
 this.fbl = fbl;  
 notifyDataSetChanged();  
 }  
  
 @Override  
 public int getCount() { return fbl == null ? 0 : fbl.size(); }  
  
 @Override  
 public FoodBean getItem(int position) { return fbl == null ? null : fbl.get(position); }  
  
 @Override  
 public long getItemId(int position) { return position; }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 ViewHolder vh = null;  
 *//复用convertView* if (convertView == null) {  
 vh = new ViewHolder();  
 convertView = LayoutInflater.*from*(context).inflate(R.layout.*menu\_item*, null);  
 vh.tv\_food\_name = (TextView) convertView.findViewById(R.id.*tv\_food\_name*);  
 vh.tv\_taste = (TextView) convertView.findViewById(R.id.*tv\_taste*);  
 vh.tv\_sale\_num = (TextView) convertView.findViewById(R.id.*tv\_sale\_num*);  
 vh.tv\_price = (TextView) convertView.findViewById(R.id.*tv\_price*);  
 vh.btn\_add\_car = (Button) convertView.findViewById(R.id.*btn\_add\_car*);  
 vh.iv\_food\_pic = (ImageView) convertView.findViewById(R.id.*iv\_food\_pic*);  
 convertView.setTag(vh);  
 } else {  
 vh = (ViewHolder) convertView.getTag();  
 }  
 *//获取position对应的Item的数据对象* FoodBean bean = getItem(position);  
 if (bean != null) {  
 vh.tv\_food\_name.setText(bean.getFoodName());  
 vh.tv\_taste.setText(bean.getTaste());  
 vh.tv\_sale\_num.setText("月售" + bean.getSaleNum());  
 vh.tv\_price.setText("￥"+bean.getPrice());  
 Glide.*with*(context)  
 .load(*WEB\_SITE* +bean.getFoodPic())  
 .error(R.mipmap.*ic\_launcher*)  
 .into(vh.iv\_food\_pic);  
 }  
 *//每个Item的点击事件* convertView.setOnClickListener(v -> {  
 *//跳转到菜品详情界面* if (bean == null)return;  
 Intent intent = new Intent(context, FoodActivity.class);  
 *//把菜品的详细信息传递到菜品详情界面* intent.putExtra("food", bean);  
 context.startActivity(intent);  
 });  
 vh.btn\_add\_car.setOnClickListener(view -> { *//加入购物车按钮的点击事件* onSelectListener.onSelectAddCar(position);  
 });  
 return convertView;  
 }  
  
 static class ViewHolder {  
 public TextView tv\_food\_name, tv\_taste, tv\_sale\_num, tv\_price;  
 public Button btn\_add\_car;  
 public ImageView iv\_food\_pic;  
 }  
 public interface OnSelectListener {  
 void onSelectAddCar (int position); *//处理加入购物车按钮的方法* }  
}

### 编写店铺列表适配器

由于店铺详情界面中的购物车列表是用 ListView 控件展示的，所以需要创建一个数据适配器 CarAdapter 对 ListView 控件进行数据适配。编写购物车列表适配器的具体步骤如下。

创建购物车列表适配器 CarAdapter。在包中创建一个适配器 CarAdapter，在该适配器中重写 getCount() 方法、getItem() 方法、getItemId() 方法和 getView() 方法。

创建 ViewHolder 类。在 CarAdapter 中创建一个 ViewHolder 类，该类主要用于创建购物车列表条目界面上的控件对象，当购物车列表快速滑动时，该类可以快速为界面控件设置值，而不必每次都重新创建很多控件对象，这样可以提高程序的性能。

创建 OnSelectListener 接口。当点击购物车列表界面的添加或减少菜品数量的按钮时，购物车中菜品的数量会随之变化，该数量的变化需要在 ShopDetailActivity 中进行，因此需要在 CarAdapter 中创建一个 OnSelectListener 接口，在该接口中创建 onSelectAdd() 方法与 onSelectMis() 方法，分别用于处理增加或减少菜品数量按钮的点击事件。

完整代码如下。

package com.hujunyao.cumtorder.adapter;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.hujunyao.cumtorder.R;  
import com.hujunyao.cumtorder.bean.FoodBean;  
  
import java.math.BigDecimal;  
import java.util.List;  
  
  
public class CarAdapter extends BaseAdapter {  
 private Context context;  
 private List<FoodBean> fbl;  
 private OnSelectListener onSelectListener;  
 public CarAdapter(Context context, OnSelectListener onSelectListener) {  
 this.context = context;  
 this.onSelectListener=onSelectListener;  
 }  
  
 public void setData(List<FoodBean> fbl) {  
 this.fbl = fbl;  
 notifyDataSetChanged();  
 }  
  
 @Override  
 public int getCount() { return fbl == null ? 0 : fbl.size(); }  
  
 @Override  
 public FoodBean getItem(int position) { return fbl == null ? null : fbl.get(position); }  
  
 @Override  
 public long getItemId(int position) { return position; }  
  
 @Override  
 public View getView(final int position, View convertView, ViewGroup parent) {  
 final ViewHolder vh;  
 *//复用convertView* if (convertView == null) {  
 vh = new ViewHolder();  
 convertView = LayoutInflater.*from*(context).inflate(R.layout.*car\_item*, null);  
 vh.tv\_food\_name = (TextView) convertView.findViewById(R.id.*tv\_food\_name*);  
 vh.tv\_food\_count = (TextView) convertView.findViewById(R.id.*tv\_food\_count*);  
 vh.tv\_food\_price = (TextView) convertView.findViewById(R.id.*tv\_food\_price*);  
 vh.iv\_add = (ImageView) convertView.findViewById(R.id.*iv\_add*);  
 vh.iv\_minus = (ImageView) convertView.findViewById(R.id.*iv\_minus*);  
 convertView.setTag(vh);  
 } else {  
 vh = (ViewHolder) convertView.getTag();  
 }  
 *//获取position对应的Item的数据对象* final FoodBean bean = getItem(position);  
 if (bean != null) {  
 vh.tv\_food\_name.setText(bean.getFoodName());  
 vh.tv\_food\_count.setText(bean.getCount()+"");  
 BigDecimal count=BigDecimal.*valueOf*(bean.getCount());  
 vh.tv\_food\_price.setText("￥" + bean.getPrice().multiply(count));  
 }  
 vh.iv\_add.setOnClickListener(view ->  
 onSelectListener.onSelectAdd(position,vh.tv\_food\_count,vh.tv\_food\_price));  
 vh.iv\_minus.setOnClickListener(view  
 -> onSelectListener.onSelectMis(position,vh.tv\_food\_count,vh.tv\_food\_price));  
 return convertView;  
 }  
 class ViewHolder {  
 public TextView tv\_food\_name, tv\_food\_count,tv\_food\_price;  
 public ImageView iv\_add,iv\_minus;  
 }  
 public interface OnSelectListener {  
 void onSelectAdd(int position,TextView tv\_food\_price,TextView tv\_food\_count);  
 void onSelectMis(int position,TextView tv\_food\_price,TextView tv\_food\_count);  
 }  
}

### 编写店铺列表适配器

店铺详情界面主要是展示店铺信息、菜单列表信息以及购物车信息，其中在菜单列表中可以点击“加入购物车”按钮，将菜品添加到购物车中。此时点击购物车图片会从界面底部弹出一个购物车列表，该列表显示的是购物车中添加的菜品信息，这些菜品信息在列表中可以进行增加和删除。点击购物车列表右上角的“清空”按钮，程序会弹出一个确认清空购物车的对话框，点击对话框中的“清空”按钮会清空购物车中的数据。修改 ShopDetailActivity 类代码，完整代码如下。

package com.hujunyao.cumtorder.activity;  
import android.annotation.SuppressLint;  
import android.app.Dialog;  
import android.content.Intent;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.os.Handler;  
import android.os.Message;  
import android.view.View;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
import com.bumptech.glide.Glide;  
import com.hujunyao.cumtorder.R;  
import com.hujunyao.cumtorder.adapter.CarAdapter;  
import com.hujunyao.cumtorder.adapter.MenuAdapter;  
import com.hujunyao.cumtorder.bean.FoodBean;  
import com.hujunyao.cumtorder.bean.ShopBean;  
import com.hujunyao.cumtorder.utils.Constant;  
import java.io.Serializable;  
import java.math.BigDecimal;  
import java.util.ArrayList;  
import java.util.Iterator;  
import java.util.List;  
  
public class ShopDetailActivity extends AppCompatActivity implements View.OnClickListener{  
 private ShopBean bean;  
 private TextView tv\_shop\_name, tv\_time, tv\_notice, tv\_title, tv\_back,  
 tv\_settle\_accounts, tv\_count, tv\_money, tv\_distribution\_cost,  
 tv\_not\_enough, tv\_clear;  
 private ImageView iv\_shop\_pic, iv\_shop\_car;  
 private ListView lv\_list, lv\_car;  
 public static final int *MSG\_COUNT\_OK* = 1;*// 获取购物车中商品的数量* private int totalCount = 0;  
 private BigDecimal totalMoney; *//购物车中菜品的总价格* private List<FoodBean> carFoodList; *//购物车中的菜品数据* private MenuAdapter adapter;  
 private CarAdapter carAdapter;  
 private RelativeLayout rl\_car\_list;  
 private MHandler mHandler;  
 private ImageView iv\_clear;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_shop\_detail*);  
 bean = (ShopBean) getIntent().getSerializableExtra("shop");  
 if (bean == null) { return; }  
 mHandler = new MHandler();  
 totalMoney = new BigDecimal("0.0");*//初始化变量totalMoney* carFoodList = new ArrayList<>(); *//初始化集合carFoodList* initView(); *//初始化界面控件* initAdapter(); *//初始化adapter* setData(); *//设置界面数据* }  
 @SuppressLint("ClickableViewAccessibility")  
 private void initView() {  
 tv\_back = (TextView) findViewById(R.id.*tv\_back*);  
 tv\_title = (TextView) findViewById(R.id.*tv\_title*);  
 tv\_title.setText("店铺详情");  
 tv\_shop\_name = (TextView) findViewById(R.id.*tv\_shop\_name*);  
 tv\_time = (TextView) findViewById(R.id.*tv\_time*);  
 tv\_notice = (TextView) findViewById(R.id.*tv\_notice*);  
 iv\_shop\_pic = (ImageView) findViewById(R.id.*iv\_shop\_pic*);  
 lv\_list = (ListView) findViewById(R.id.*lv\_list*);  
 tv\_settle\_accounts = (TextView) findViewById(R.id.*tv\_settle\_accounts*);  
 tv\_distribution\_cost = (TextView) findViewById(R.id.*tv\_distribution\_cost*);  
 tv\_count = (TextView) findViewById(R.id.*tv\_count*);  
 iv\_shop\_car = (ImageView) findViewById(R.id.*iv\_shop\_car*);  
 tv\_money = (TextView) findViewById(R.id.*tv\_money*);  
 tv\_not\_enough = (TextView) findViewById(R.id.*tv\_not\_enough*);  
 tv\_clear = (TextView) findViewById(R.id.*tv\_clear*);  
 iv\_clear = (ImageView) findViewById(R.id.*iv\_clear*);  
 lv\_car = (ListView) findViewById(R.id.*lv\_car*);  
 rl\_car\_list = findViewById(R.id.*rl\_car\_list*);  
 *//点击购物车列表界面外的其他部分会隐藏购物车列表界面* rl\_car\_list.setOnTouchListener((v, event) -> {  
 if (rl\_car\_list.getVisibility() == View.*VISIBLE*) {  
 rl\_car\_list.setVisibility(View.*GONE*);  
 }  
 return false;  
 });  
 *//设置返回键、去结算按钮、购物车图片、清空购物车按钮的点击监听事件* tv\_back.setOnClickListener(this);  
 tv\_settle\_accounts.setOnClickListener(this);  
 iv\_shop\_car.setOnClickListener(this);  
 tv\_clear.setOnClickListener(this);  
 iv\_clear.setOnClickListener(this);  
 }  
 private void initAdapter(){  
 carAdapter = new CarAdapter(this, new CarAdapter.OnSelectListener() {  
 @SuppressLint("SetTextI18n")  
 @Override  
 public void onSelectAdd(int position, TextView tv\_food\_count, TextView tv\_food\_price) {  
 *//添加菜品到购物车中* FoodBean bean = carFoodList.get(position); *//获取当前菜品对象* tv\_food\_count.setText(bean.getCount() + 1 + ""); *//设置该菜品在购物车中的数量* BigDecimal count = BigDecimal.*valueOf*(bean.getCount() + 1);  
 tv\_food\_price.setText("￥" + bean.getPrice().multiply(count));*//菜品总价格* bean.setCount(bean.getCount() + 1);*//将当前菜品在购物车中的数量设置给菜品对象* Iterator<FoodBean> iterator = carFoodList.iterator();  
 while (iterator.hasNext()) {*//遍历购物车中的数据* FoodBean food = iterator.next();  
 if (food.getFoodId() == bean.getFoodId()) {*//找到当前菜品* iterator.remove(); *//删除购物车中当前菜品的旧数据* }  
 }  
 carFoodList.add(position, bean); *//将当前菜品的最新数据添加到购物车数据集合中* totalCount = totalCount + 1; *//购物车中菜品的总数量+1  
 //购物车中菜品的总价格+当前菜品价格* totalMoney = totalMoney.add(bean.getPrice());  
 carDataMsg();*//将购物车中菜品的总数量和总价格通过Handler传递到主线程中* }  
 @SuppressLint("SetTextI18n")  
 @Override  
 public void onSelectMis(int position, TextView tv\_food\_count, TextView tv\_food\_price) {  
 FoodBean bean = carFoodList.get(position); *//获取当前菜品对象* tv\_food\_count.setText(bean.getCount() - 1 + "");*//设置当前菜品的数量* BigDecimal count = BigDecimal.*valueOf*(bean.getCount() - 1);  
 *//设置当前菜品总价格，菜品价格=菜品单价\*菜品数量* tv\_food\_price.setText("￥" + bean.getPrice().multiply(count));  
 minusCarData(bean, position);*//删除购物车中的菜品* }  
 });  
 adapter = new MenuAdapter(this, position -> {  
 *//点击加入购物车按钮将菜添加到购物车中* FoodBean fb = bean.getFoodList().get(position);  
 fb.setCount(fb.getCount() + 1);  
 Iterator<FoodBean> iterator = carFoodList.iterator();  
 while (iterator.hasNext()) {  
 FoodBean food = iterator.next();  
 if (food.getFoodId() == fb.getFoodId()) {  
 iterator.remove();  
 }  
 }  
 carFoodList.add(fb);  
 totalCount = totalCount + 1;  
 totalMoney = totalMoney.add(fb.getPrice());  
 carDataMsg();  
 });  
 lv\_list.setAdapter(adapter);  
 }  
  
 private void minusCarData(FoodBean bean, int position) {  
 int count = bean.getCount() - 1; *//将该菜品的数量减1* bean.setCount(count); *//将减后的数量设置到菜品对象中* Iterator<FoodBean> iterator = carFoodList.iterator();  
 while (iterator.hasNext()) { *//遍历购物车中的菜* FoodBean food = iterator.next();  
 if (food.getFoodId() == bean.getFoodId()) {*//找到购物车中当前菜的Id* iterator.remove(); *//删除存放的菜* }  
 }  
 *//如果当前菜品的数量减1之后大于0，则将当前菜品添加到购物车集合中* if (count > 0) carFoodList.add(position, bean);  
 else carAdapter.notifyDataSetChanged();  
 totalCount = totalCount - 1; *//购物车中菜品的数量减1  
 //购物车中的总价钱=总价钱-当前菜品的价格* totalMoney = totalMoney.subtract(bean.getPrice());  
 carDataMsg(); *//调用该方法更新购物车中的数据* }  
  
 @SuppressLint("SetTextI18n")  
 private void setData() {  
 if (bean == null) return;  
 tv\_shop\_name.setText(bean.getShopName()); *//设置店铺名称* tv\_time.setText("大约用时"+bean.getTime()+"分钟"); *//设置配送时间* tv\_notice.setText(bean.getShopNotice()); *//设置店铺公告* tv\_not\_enough.setText("￥" + bean.getOfferPrice() + "起送"); *//设置起送价格* Glide.*with*(this)  
 .load(Constant.*WEB\_SITE* + bean.getShopPic())  
 .error(R.mipmap.*ic\_launcher*)  
 .into(iv\_shop\_pic); *//设置店铺图片* adapter.setData(bean.getFoodList());*//将菜单列表数据传递到adapter中* }  
  
 private void dialogClear() {  
 *//创建一个对话框Dialog* final Dialog dialog = new Dialog(ShopDetailActivity.this, R.style.*Dialog\_Style*);  
 dialog.setContentView(R.layout.*dialog\_clear*); *//将布局文件加载到对话框中* dialog.show(); *//显示对话框* TextView tv\_clear = dialog.findViewById(R.id.*tv\_clear*); *//获取对话框清除按钮* TextView tv\_cancel = dialog.findViewById(R.id.*tv\_cancel*);*//获取对话框取消按钮* tv\_cancel.setOnClickListener(view -> {  
 dialog.dismiss();*//关闭对话框* });  
 tv\_clear.setOnClickListener(view -> {  
 if (carFoodList == null) return;  
 for (FoodBean bean : carFoodList) {  
 bean.setCount(0);*//设置购物车中所有菜品的数量为0* }  
 carFoodList.clear();*//清空购物车中的数据* carAdapter.notifyDataSetChanged(); *//更新界面* totalCount = 0; *//购物车中菜品的数量设置为0* totalMoney = BigDecimal.*valueOf*(0.0);*//总价格设置为0* carDataMsg(); *//通过Handler更新购物车中菜品的数量和总价格* dialog.dismiss(); *//关闭对话框* });  
 }  
  
 @SuppressLint("NonConstantResourceId")  
 @Override  
 public void onClick(View v) {  
 switch (v.getId()){  
 case R.id.*tv\_back*: *//返回按钮的点击事件* finish();  
 break;  
 case R.id.*tv\_settle\_accounts*: *//去结算按钮的点击事件  
 //跳转到订单界面* if (totalCount > 0) {  
 Intent intent = new Intent(ShopDetailActivity.this, OrderActivity.class);  
 intent.putExtra("carFoodList", (Serializable) carFoodList);  
 intent.putExtra("totalMoney", totalMoney + "");  
 intent.putExtra("distributionCost", bean.getDistributionCost() + "");  
 startActivity(intent);  
 }  
 break;  
 case R.id.*iv\_shop\_car*: *//购物车的点击事件* if (totalCount <= 0) return;  
 if (rl\_car\_list != null) {  
 if (rl\_car\_list.getVisibility() == View.*VISIBLE*) {  
 rl\_car\_list.setVisibility(View.*GONE*);  
 } else {  
 rl\_car\_list.setVisibility(View.*VISIBLE*);  
 *//创建一个从底部滑出的动画* Animation animation = AnimationUtils.*loadAnimation*(  
 ShopDetailActivity.this, R.anim.*slide\_bottom\_to\_top*);  
 rl\_car\_list.startAnimation(animation);*//将动画加载到购物车列表界面* }  
 }  
 carAdapter.setData(carFoodList);  
 lv\_car.setAdapter(carAdapter);  
 break;  
 case R.id.*tv\_clear*:*//清空按钮的点击事件* case R.id.*iv\_clear*:  
 dialogClear(); *//弹出确认清空购物车的对话框* break;  
 }  
 }  
  
 private void carDataMsg() {  
 Message msg = Message.*obtain*();  
 msg.what = *MSG\_COUNT\_OK*;  
 Bundle bundle = new Bundle();*//创建一个Bundler对象  
 //将购物车中的菜品数量和价格放入Bundler对象中* bundle.putString("totalCount", totalCount + "");  
 bundle.putString("totalMoney", totalMoney + "");  
 msg.setData(bundle); *//将Bundler对象放入Message对象* mHandler.sendMessage(msg); *//将Message对象传递到MHandler类* }  
  
 @SuppressLint("HandlerLeak")  
 class MHandler extends Handler {  
 @SuppressLint("SetTextI18n")  
 @Override  
 public void dispatchMessage(Message msg) {  
 super.dispatchMessage(msg);  
 if (msg.what == *MSG\_COUNT\_OK*) {  
 Bundle bundle = msg.getData();  
 String count = bundle.getString("totalCount", "");  
 String money = bundle.getString("totalMoney", "");  
 if (Integer.*parseInt*(count) > 0) {*//如果购物车中有菜品* iv\_shop\_car.setImageResource(R.drawable.*shop\_car*);  
 tv\_count.setVisibility(View.*VISIBLE*);  
 tv\_distribution\_cost.setVisibility(View.*VISIBLE*);  
 tv\_money.setTextColor(Color.*parseColor*("#ffffff"));  
 tv\_money.getPaint().setFakeBoldText(true);*//加粗字体* tv\_money.setText("￥" + money);*//设置购物车中菜品总价格* tv\_count.setText(count); *//设置购物车中菜品总数量* tv\_distribution\_cost.setText("另需配送费￥" + bean.getDistributionCost());  
 *//将变量money的类型转换为BigDecimal类型* BigDecimal bdm = new BigDecimal(money);  
 *//总价格money与起送价格对比* int result = bdm.compareTo(bean.getOfferPrice());  
 if (result < 0) { *//总价格<起送价格* tv\_settle\_accounts.setVisibility(View.*GONE*);*//隐藏去结算按钮* tv\_not\_enough.setVisibility(View.*VISIBLE*); *//显示差价文本  
 //差价=起送价格-总价格* BigDecimal m = bean.getOfferPrice().subtract(bdm);  
 tv\_not\_enough.setText("还差￥" + m + "起送");  
 } else { *//总价格>=起送价格  
 //显示去结算按钮* tv\_settle\_accounts.setVisibility(View.*VISIBLE*);  
 tv\_not\_enough.setVisibility(View.*GONE*); *//隐藏差价文本* }  
 } else { *//如果购物车中没有菜品* if (rl\_car\_list.getVisibility() == View.*VISIBLE*) {  
 rl\_car\_list.setVisibility(View.*GONE*); *//隐藏购物车列表* } else {  
 rl\_car\_list.setVisibility(View.*VISIBLE*);*//显示购物车列表* }  
 iv\_shop\_car.setImageResource(R.drawable.*shop\_car\_empty*);  
 tv\_settle\_accounts.setVisibility(View.*GONE*);*//隐藏去结算按钮* tv\_not\_enough.setVisibility(View.*VISIBLE*); *//显示差价文本* tv\_not\_enough.setText("￥" + bean.getOfferPrice() + "起送");  
 tv\_count.setVisibility(View.*GONE*);*//隐藏购物中的菜品数量控件* tv\_distribution\_cost.setVisibility(View.*GONE*);*//隐藏配送费用*tv\_money.setTextColor(getResources().getColor(R.color.*light\_gray*, null));  
 tv\_money.setText("未选购商品");  
 }  
 }  
 }  
 }  
}

## 菜品详情功能业务实现

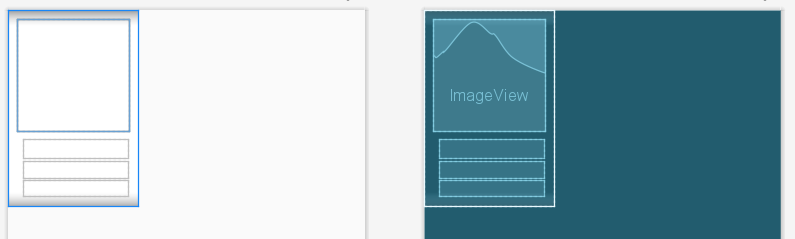
点击菜单列表的条目，程序会跳转到菜品详情界面，该界面主要用于展示菜品的名称、月售数量和价格等信息。菜品详情界面中的数据是从店铺详情界面传递过来的。接下来本节将针对菜品详情功能业务的实现进行详细讲解。

### 搭建菜品详情界面布局

创建 FoodActivity。指定布局文件名为 activity\_food.xml。放置3个 TextView 控件、1个 ImageView 控件。在 styles.xml 文件中创建对话框样式 Theme.ActivityDialogStyle。（已导入）完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:background="@android:color/white"  
 android:padding="10dp"  
 android:orientation="vertical">  
 <ImageView  
 android:id="@+id/iv\_food\_pic"  
 android:layout\_width="130dp"  
 android:layout\_height="130dp"  
 android:layout\_gravity="center\_horizontal"  
 android:scaleType="fitXY" />  
 <TextView  
 android:id="@+id/tv\_food\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:textColor="@android:color/black"  
 android:textSize="16sp" />  
 <TextView  
 android:id="@+id/tv\_sale\_num"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="4dp"  
 android:textColor="@color/color\_gray"  
 android:textSize="14sp" />  
 <TextView  
 android:id="@+id/tv\_price"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="4dp"  
 android:textColor="@color/price\_red"  
 android:textSize="14sp" />  
</LinearLayout>

activity\_food.xml 界面如下。



### 实现菜品界面显示功能

菜品详情界面的数据是从店铺详情界面传递过来的，该界面的逻辑代码相对比较简单，主要是获取传递过来的菜品数据，并将数据显示到界面上。实现菜品界面显示功能的具体步骤如下。

获取界面控件。在FoodActivity中创建初始化界面控件的方法initView()。

设置界面数据。FoodActivity中创建一个setData()方法，该方法用于将数据设置到菜品详情界面的控件上。

修改MenuAdapter.java文件。在MenuAdapter中的getView()方法中的注释“//跳转到菜品详情界面”下方添加跳转到菜品详情界面的逻辑代码。

package com.hujunyao.cumtorder.activity;  
  
import android.os.Bundle;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.bumptech.glide.Glide;  
import com.hujunyao.cumtorder.R;  
import com.hujunyao.cumtorder.bean.FoodBean;  
import com.hujunyao.cumtorder.utils.Constant;  
  
public class FoodActivity extends AppCompatActivity {  
 private FoodBean bean;  
 private TextView tv\_food\_name, tv\_sale\_num, tv\_price;  
 private ImageView iv\_food\_pic;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_food*);  
 *//从店铺详情界面传递过来的菜的数据* bean = (FoodBean) getIntent().getSerializableExtra("food");  
  
 initView();  
 setData();  
 }  
  
 private void initView() {  
 tv\_food\_name = (TextView) findViewById(R.id.*tv\_food\_name*);  
 tv\_sale\_num = (TextView) findViewById(R.id.*tv\_sale\_num*);  
 tv\_price = (TextView) findViewById(R.id.*tv\_price*);  
 iv\_food\_pic = (ImageView) findViewById(R.id.*iv\_food\_pic*);  
 }  
 */\*\*  
 \* 设置界面数据  
 \*/* private void setData() {  
 if (bean == null) return;  
 tv\_food\_name.setText(bean.getFoodName());  
 tv\_sale\_num.setText("月售" + bean.getSaleNum());  
 tv\_price.setText("￥" + bean.getPrice());  
 Glide.with(this)  
 .load(Constant.WEB\_SITE + bean.getFoodPic())  
 .error(R.mipmap.ic\_launcher)  
 .into(iv\_food\_pic);  
 }  
}

## 订单功能业务实现

在店铺详情界面，点击“去结算”按钮，程序会跳转到订单界面，订单界面主要展示的是收货地址、订单列表、小计、配送费以及订单总价与“去支付”按钮，该界面的数据是从店铺详情界面传递过来的。点击“去支付”按钮，程序会弹出一个二维码支付界面供用户付款。接下来本节将针对订单功能业务的实现进行详细讲解。

### 搭建订单界面布局

创建 OrderActivity。指定布局文件名为 activity\_order.xml。在 activity\_order.xml 中引入 order\_head.xml 与 payment.xml 文件。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.hujunyao.cumtorder">  
 <uses-permission android:name="android.permission.INTERNET" />  
 <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/app\_icon"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/app\_icon"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.CumtOrder"  
 android:usesCleartextTraffic="true">  
 <activity  
 android:name=".activity.SplashActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <activity android:name=".activity.LoginActivity" />  
 <activity android:name=".activity.RegisterActivity" />  
 <activity android:name=".activity.ShopActivity" />  
 <activity android:name=".activity.ShopDetailActivity" />  
 <activity  
 android:name=".activity.FoodActivity"  
 android:theme="@style/Theme.ActivityDialogStyle" />  
 <activity android:name=".activity.OrderActivity" />  
 </application>  
</manifest>

创建 order\_head.xml 文件，完整代码如下。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/type\_gray"  
 android:orientation="vertical">  
 <include layout="@layout/title\_bar" />  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="15dp"  
 android:background="@android:color/white">  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:text="收货地址"  
 android:textColor="@color/color\_gray"  
 android:textSize="16sp"  
 android:layout\_weight="2"/>  
 <EditText  
 android:id="@+id/et\_address"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:background="@android:color/white"  
 android:hint="如：矿大南湖校区计算机学院"  
 android:scrollbars="horizontal"  
 android:singleLine="true"  
 android:textColor="@android:color/black"  
 android:textSize="16sp"  
 android:layout\_weight="7"/>  
 <ImageView  
 android:id="@+id/iv\_address"  
 android:layout\_width="0dp"  
 android:layout\_height="20dp"  
 android:src="@drawable/car\_add"  
 android:layout\_gravity="center\_vertical"  
 android:layout\_weight="1"  
 />  
 </LinearLayout>  
 <ListView  
 android:id="@+id/lv\_order"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:background="@android:color/white" >  
 </ListView>  
 <View  
 android:layout\_width="match\_parent"  
 android:layout\_height="1dp"  
 android:background="@color/type\_gray" />  
 <RelativeLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@android:color/white">  
 <TextView  
 android:id="@+id/tv\_cost"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:padding="10dp"  
 android:textColor="@color/price\_red"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="2dp"  
 android:layout\_toStartOf="@id/tv\_cost"  
 android:padding="10dp"  
 android:text="小计"  
 android:textColor="@android:color/black"  
 android:textSize="14sp" />  
 <TextView  
 android:id="@+id/tv\_distribution\_cost"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_below="@id/tv\_cost"  
 android:padding="10dp"  
 android:textColor="@android:color/black"  
 android:textSize="14sp"  
 android:textStyle="bold" />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/tv\_cost"  
 android:layout\_toStartOf="@id/tv\_distribution\_cost"  
 android:padding="10dp"  
 android:text="配送费"  
 android:textColor="@android:color/black"  
 android:textSize="14sp" />  
 </RelativeLayout>  
</LinearLayout>

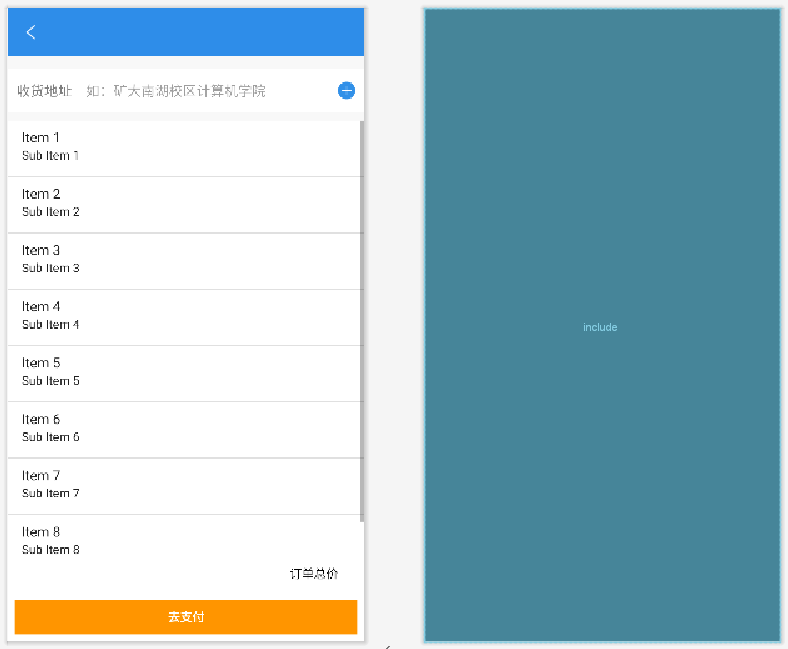
创建 payment.xml 文件，完整代码如下。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@android:color/transparent"  
 android:gravity="bottom">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@android:color/white">  
  
 <TextView  
 android:id="@+id/tv\_total\_cost"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:padding="10dp"  
 android:textColor="@color/price\_red"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="2dp"  
 android:layout\_toStartOf="@id/tv\_total\_cost"  
 android:padding="10dp"  
 android:text="订单总价"  
 android:textColor="@android:color/black"  
 android:textSize="14sp" />  
  
 <TextView  
 android:id="@+id/tv\_payment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"  
 android:layout\_below="@id/tv\_total\_cost"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:background="@drawable/payment\_bg\_selector"  
 android:gravity="center"  
 android:text="去支付"  
 android:textColor="@android:color/white"  
 android:textStyle="bold" />  
 </RelativeLayout>  
</RelativeLayout>

创建背景选择器 payment\_bg\_selector.xml，完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<selector xmlns:android="http://schemas.android.com/apk/res/android">  
 <item android:drawable="@color/account\_selected\_color" android:state\_pressed="true"/>  
 <item android:drawable="@color/account\_color"/>  
</selector>

activity\_order.xml 界面如下。

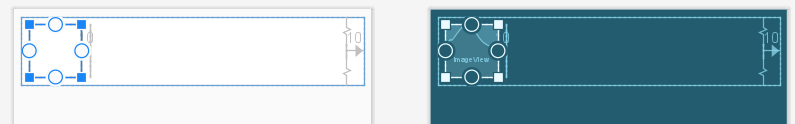


### 搭建订单列表条目界面布局

由于订单界面中使用 ListView 控件展示订单列表信息，所以需要创建一个该列表的条目界面。在条目界面中需要展示菜品的名称、数量以及总价信息。创建布局文件 order\_item.xml。放置1个ImageView控件、3个TextView控件。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginTop="8dp"  
 android:background="@drawable/menu\_item\_bg\_selector"  
 android:padding="10dp">  
 <ImageView  
 android:id="@+id/iv\_food\_pic"  
 android:layout\_width="60dp"  
 android:layout\_height="60dp"  
 android:layout\_alignParentLeft="true"  
 android:scaleType="fitXY" />  
 <LinearLayout  
 android:id="@+id/ll\_info"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="60dp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_toRightOf="@id/iv\_food\_pic"  
 android:gravity="center"  
 android:orientation="vertical">  
 <TextView  
 android:id="@+id/tv\_food\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="@android:color/black"  
 android:textSize="14sp" />  
 <TextView  
 android:id="@+id/tv\_count"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:textColor="@android:color/black"  
 android:textSize="12sp" />  
 </LinearLayout>  
 <TextView  
 android:id="@+id/tv\_money"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_centerVertical="true"  
 android:layout\_marginEnd="10dp"  
 android:textColor="@android:color/black"  
 android:textSize="12sp" />  
</RelativeLayout>

order\_item.xml 界面如下。

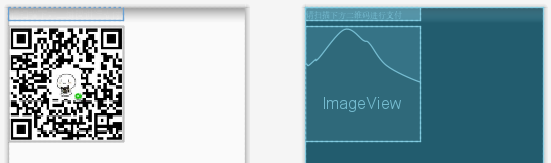


### 搭建支付界面布局

创建布局文件 qr\_code.xml。放置1个 TextView 控件、1个 ImageView 控件。完整代码如下。

<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
 <TextView  
 android:layout\_width="200dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 android:text="请扫描下方二维码进行支付"  
 android:gravity="center"  
 android:textColor="@android:color/white"  
 android:textSize="16sp" />  
 <ImageView  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:scaleType="fitXY"  
 android:src="@drawable/wxpay" />  
</LinearLayout>

qr\_code.xml 界面如下。



### 编写订单列表适配器

订单界面的订单列表信息是用 ListView 控件展示的，所以需要创建一个数据适配器 OrderAdapter 对 ListView 控件进行数据适配。编写订单列表适配器的具体步骤如下。

创建订单列表适配器 OrderAdapter。创建一个适配器 OrderAdapter，并在该适配器中重写 getCount() 方法、getItem() 方法、getItemId()方法和 getView() 方法。

创建 ViewHolder 类。在 OrderAdapter 中创建一个 ViewHolder 类，该类主要用于创建订单列表条目上的控件对象，当订单列表快速滑动时，该类可以快速为界面控件设置值，而不必每次都重新创建很多控件对象，这样可以提高程序的性能。

完整代码如下。

package com.hujunyao.cumtorder.adapter;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.BaseAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.bumptech.glide.Glide;  
import com.hujunyao.cumtorder.R;  
import com.hujunyao.cumtorder.bean.FoodBean;  
import com.hujunyao.cumtorder.utils.Constant;  
  
import java.math.BigDecimal;  
import java.util.List;  
  
  
public class OrderAdapter extends BaseAdapter {  
 private Context context;  
 private List<FoodBean> fbl;  
 public OrderAdapter(Context context) {  
 this.context = context;  
 }  
  
 public void setData(List<FoodBean> fbl) {  
 this.fbl = fbl;  
 notifyDataSetChanged();  
 }  
  
 @Override  
 public int getCount() { return fbl == null ? 0 : fbl.size(); }  
  
 @Override  
 public FoodBean getItem(int position) { return fbl == null ? null : fbl.get(position); }  
  
 @Override  
 public long getItemId(int position) { return position; }  
  
 @Override  
 public View getView(final int position, View convertView, ViewGroup parent) {  
 final ViewHolder vh;  
 *//复用convertView* if (convertView == null) {  
 vh = new ViewHolder();  
 convertView = LayoutInflater.*from*(context).inflate(R.layout.*order\_item*, null);  
 vh.tv\_food\_name = (TextView) convertView.findViewById(R.id.*tv\_food\_name*);  
 vh.tv\_count = (TextView) convertView.findViewById(R.id.*tv\_count*);  
 vh.tv\_money = (TextView) convertView.findViewById(R.id.*tv\_money*);  
 vh.iv\_food\_pic = (ImageView) convertView.findViewById(R.id.*iv\_food\_pic*);  
 convertView.setTag(vh);  
 } else {  
 vh = (ViewHolder) convertView.getTag();  
 }  
 *//获取position对应的Item的数据对象* final FoodBean bean = getItem(position);  
 if (bean != null) {  
 vh.tv\_food\_name.setText(bean.getFoodName());  
 vh.tv\_count.setText("×"+bean.getCount());  
 vh.tv\_money.setText("￥"+ bean.getPrice().  
 multiply(BigDecimal.*valueOf*(bean.getCount())));  
 Glide.*with*(context)  
 .load(Constant.*WEB\_SITE* + bean.getFoodPic())  
 .error(R.mipmap.*ic\_launcher*)  
 .into(vh.iv\_food\_pic);  
 }  
 return convertView;  
 }  
 class ViewHolder {  
 public TextView tv\_food\_name, tv\_count, tv\_money;  
 public ImageView iv\_food\_pic;  
 }  
}

### 实现订单显示与支付功能

订单界面的数据是从店铺详情界面传递过来的，该界面的逻辑代码相对比较简单，主要是获取传递过来的数据，并将数据显示到界面上。实现订单显示与支付功能的具体步骤如下：

获取界面控件。在 OrderActivity 中创建界面控件的初始化方法 init()，该方法用于获取订单界面所要用到的控件并实现返回键与“去支付”按钮的点击事件。

设置界面数据。在 OrderActivity 中创建一个 setData() 方法，该方法用于将数据设置到订单界面的控件上。

修改 ShopDetailActivity.java 文件。由于点击店铺详情界面的“去结算”按钮时，会跳转到订单界面，所以需要找到 ShopDetailActivity 中的 onClick() 方法，在该方法中的注释“//跳转到订单界面”下方添加跳转到订单界面的逻辑代码。

package com.hujunyao.cumtorder.activity;  
  
import android.app.Dialog;  
import android.os.Bundle;  
import android.util.Log;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.hujunyao.cumtorder.R;  
import com.hujunyao.cumtorder.adapter.OrderAdapter;  
import com.hujunyao.cumtorder.bean.FoodBean;  
  
import java.math.BigDecimal;  
import java.util.List;  
  
public class OrderActivity extends AppCompatActivity {  
 private ListView lv\_order;  
  
 private OrderAdapter adapter;  
 private List<FoodBean> carFoodList;  
 private TextView tv\_title,tv\_back,tv\_distribution\_cost,tv\_total\_cost,  
 tv\_cost,tv\_payment;  
 private BigDecimal money, distributionCost;  
 private EditText et\_address;  
 private ImageView iv\_address;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_order);  
 carFoodList = (List<FoodBean>) getIntent().getSerializableExtra("carFoodList");  
 money = new BigDecimal(getIntent().getStringExtra("totalMoney"));  
 distributionCost = new BigDecimal(getIntent().getStringExtra("distributionCost"));  
 initView();  
 setData();  
 }  
  
 private void initView(){  
 tv\_title = (TextView) findViewById(R.id.tv\_title);  
 tv\_title.setText("订单");  
 tv\_back = (TextView) findViewById(R.id.tv\_back);  
 lv\_order= (ListView) findViewById(R.id.lv\_order);  
 tv\_distribution\_cost = (TextView) findViewById(R.id.tv\_distribution\_cost);  
 tv\_total\_cost = (TextView) findViewById(R.id.tv\_total\_cost);  
 tv\_cost = (TextView) findViewById(R.id.tv\_cost);  
 tv\_payment = (TextView) findViewById(R.id.tv\_payment);  
 et\_address = (EditText) findViewById(R.id.et\_address);  
 et\_address.setHorizontallyScrolling(true);  
 iv\_address = (ImageView) findViewById(R.id.iv\_address);  
  
 *//选择地址* iv\_address.setOnClickListener(V->{  
 et\_address.setText("矿大南湖计算机学院306");  
 });  
  
 *// 返回键的点击事件* tv\_back.setOnClickListener(v -> finish());  
 tv\_payment.setOnClickListener(view -> { *//“去支付”按钮的点击事件  
// Log.v("tag", et\_address.getText().toString().trim());* if (!(et\_address.getText().toString().trim()).equals("")) {  
 Dialog dialog = new Dialog(OrderActivity.this, R.style.Dialog\_Style);  
 dialog.setContentView(R.layout.qr\_code);  
 dialog.show();  
 }  
 else {  
 Toast.makeText(this, "请输入您的地址", Toast.LENGTH\_SHORT).show();  
 }  
 });  
 }  
 */\*\*  
 \* 设置界面数据  
 \*/* private void setData() {  
 adapter=new OrderAdapter(this);  
 lv\_order.setAdapter(adapter);  
 adapter.setData(carFoodList);  
 tv\_cost.setText("￥"+money);  
 tv\_distribution\_cost.setText("￥"+distributionCost);  
 tv\_total\_cost.setText("￥"+(money.add(distributionCost)));  
 }  
}

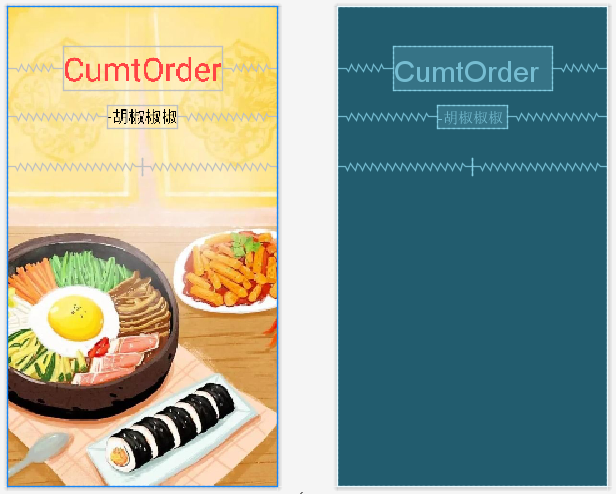
## 实现欢迎页面

### 实现欢迎页面布局

在 activity 包下新建一个类，命名为 SplashActivity。在 res/layout 下创建 xml 布局文件，命名为 activity\_splash.xml。需要引入3个 TextView，包括软件名字，作者名字，版本号。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>  
<!-- match\_parent:占据手机界面全部 wrap\_content：占据自己的那一部分-->*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/launch\_bg">  
 *<!-- 设置欢迎界面背景图片 -->  
  
 <!-- 文本控件用于显示项目版本号 -->* <TextView  
 android:id="@+id/tv\_app\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="60dp"  
 android:text="CumtOrder"  
 android:textColor="@android:color/holo\_red\_light"  
 android:textSize="50sp" />  
 <TextView  
 android:id="@+id/tv\_app\_author"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="150dp"  
 android:text="-胡椒椒椒"  
 android:textColor="@android:color/black"  
 android:textSize="25sp" />  
  
 <TextView  
 android:id="@+id/tv\_version"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="230dp"  
 android:textColor="@android:color/holo\_blue\_dark"  
 android:textSize="20sp" />  
</RelativeLayout>

activity\_splash.xml 界面如下。



### 实现欢迎页面功能

打开软件，等待三秒钟，之后自动进入登陆界面。完整代码如下。

package com.hujunyao.cumtorder.activity;  
  
import android.content.Intent;  
import android.content.pm.ActivityInfo;  
import android.content.pm.PackageInfo;  
import android.content.pm.PackageManager;  
import android.os.Bundle;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.hujunyao.cumtorder.R;  
import java.util.Timer;  
import java.util.TimerTask;  
  
public class SplashActivity extends AppCompatActivity {  
  
 private TextView tv\_version; *//新建tv\_version成员变量* @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_splash);  
 *//设置此界面为竖屏* setRequestedOrientation(ActivityInfo.SCREEN\_ORIENTATION\_UNSPECIFIED);  
 init();  
 }  
  
 private void init() {  
 *//将布局文件的tv\_version属性赋给成员变量tv\_version* tv\_version = (TextView) findViewById(R.id.tv\_version);  
 try {  
 *//获取程序包信息* PackageInfo info = getPackageManager().getPackageInfo(getPackageName(), 0);  
 *//获得版本号：AndroidManifest.xml里面的versionName* tv\_version.setText("V" + info.versionName);  
 } catch (PackageManager.NameNotFoundException e) {  
 *//如果获取不到版本号，控制台打印错误日志，System.err* e.printStackTrace();  
 *//欢迎界面返回V* tv\_version.setText("V");  
 }  
 *//利用Timer让此界面延迟3秒后跳转,timer中有一个线程，这个线程不断执行task* Timer timer = new Timer();  
 *//timertask实现runnable接口,TimerTask类表示一个在指定时间内执行的task* TimerTask task = new TimerTask() {  
 @Override  
 public void run() {  
 *//欢迎界面跳转到主界面* Intent intent = new Intent(SplashActivity.this, LoginActivity.class);  
 startActivity(intent);  
 SplashActivity.this.finish();  
 }  
 };  
 timer.schedule(task, 3000);*//设置这个task在延迟3秒之后自动执行* }  
}

## 实现注册页面

### 实现注册页面功能

创建一个类，命名为RegisterActivity。在res/layout文件夹中，创建activity\_register.xml。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/register\_bg"  
 android:orientation="vertical">  
  
 *<!-- 导入标题栏——main\_title\_bar.xml文件 -->* <include layout="@layout/title\_bar"></include>  
  
 *<!-- 注册界面头像——default\_icon.png -->* <ImageView  
 android:layout\_width="70dp"  
 android:layout\_height="70dp"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginTop="25dp"  
 android:src="@mipmap/app\_icon"  
 android:background="@drawable/icon\_bg" />  
  
 *<!-- 用户名输入框 -->* <EditText  
 android:id="@+id/et\_user\_name"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="48dp"  
 android:layout\_marginLeft="35dp"  
 android:layout\_marginRight="35dp"  
 android:layout\_marginTop="35dp"  
 android:background="@drawable/register\_user\_name\_bg"  
 android:drawableStart="@drawable/user\_name\_icon"  
 android:drawablePadding="10dp"  
 android:gravity="center\_vertical"  
 android:hint="请输入用户名"  
 android:paddingStart="8dp"  
 android:singleLine="true"  
 android:textColor="#000000"  
 android:textColorHint="#a3a3a3"  
 android:textSize="14sp"/>  
  
 *<!-- 密码输入框 -->* <EditText  
 android:id="@+id/et\_psw"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="48dp"  
 android:gravity="center\_vertical"  
 android:layout\_marginLeft="35dp"  
 android:layout\_marginRight="35dp"  
 android:background="@drawable/register\_psw\_bg"  
 android:drawableStart="@drawable/psw\_icon"  
 android:drawablePadding="10dp"  
 android:hint="请输入密码"  
 android:inputType="textPassword"  
 android:paddingStart="8dp"  
 android:singleLine="true"  
 android:textColor="#000000"  
 android:textColorHint="#a3a3a3"  
 android:textSize="14sp"/>  
  
 *<!-- 确认密码输入框 -->* <EditText  
 android:id="@+id/et\_psw\_again"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="48dp"  
 android:gravity="center\_vertical"  
 android:layout\_marginLeft="35dp"  
 android:layout\_marginRight="35dp"  
 android:background="@drawable/register\_psw\_again\_bg"  
 android:drawableStart="@drawable/psw\_icon"  
 android:drawablePadding="10dp"  
 android:hint="请再次输入密码"  
 android:inputType="textPassword"  
 android:paddingStart="8dp"  
 android:singleLine="true"  
 android:textColor="#000000"  
 android:textColorHint="#a3a3a3"  
 android:textSize="14sp"/>  
  
 *<!-- 注册按钮——使用背景选择器register\_selector.xml -->* <Button  
 android:id="@+id/btn\_register"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="40dp"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginLeft="35dp"  
 android:layout\_marginRight="35dp"  
 android:layout\_marginTop="15dp"  
 android:background="@drawable/register\_selector"  
 android:text="注 册"  
 android:textColor="@android:color/white"  
 android:textSize="18sp"/>  
</LinearLayout>

### 实现注册页面功能

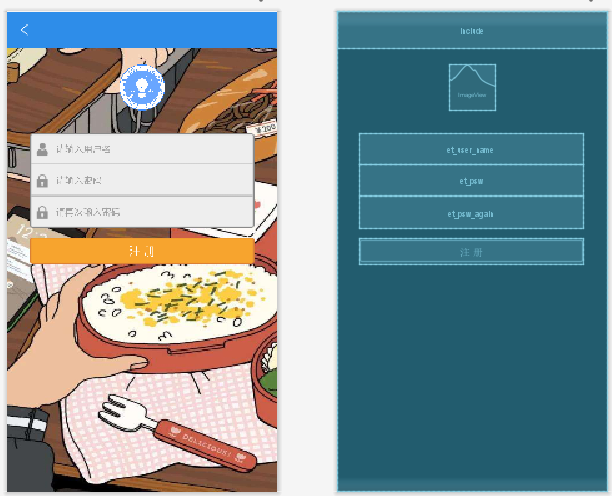
注册页面需要从登录界面进入。注册账号成功后可以进行登录。

package com.hujunyao.cumtorder.activity;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.content.pm.ActivityInfo;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.hujunyao.cumtorder.R;  
import com.hujunyao.cumtorder.utils.MD5Utils;  
  
  
public class RegisterActivity extends AppCompatActivity {  
 *//标题栏* private TextView tv\_title;  
 *//返回按钮* private TextView tv\_back;  
 *//注册按钮* private Button btn\_register;  
 *//账号、密码、再次输入密码的控件* private EditText et\_user\_name,et\_psw,et\_psw\_again;  
 *//账号、密码、再次输入密码的控件的获取值* private String userName,psw,pswAgain;  
 *//标题布局* private RelativeLayout rl\_rirle\_bar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 *//设置页面布局* setContentView(R.layout.activity\_register);  
 *//设置界面为竖屏* setRequestedOrientation(ActivityInfo.SCREEN\_ORIENTATION\_UNSPECIFIED);  
 init();  
 }  
  
 private void init(){  
 *//main\_title\_bar.xml页面布局文件获取对应的UI控件* tv\_title = (TextView)findViewById(R.id.tv\_title);  
 tv\_title.setText("注册");  
 tv\_back = (TextView)findViewById(R.id.tv\_back);  
 tv\_back.setVisibility(View.GONE);  
 rl\_rirle\_bar = (RelativeLayout)findViewById(R.id.title\_bar);  
 rl\_rirle\_bar.setBackgroundColor(Color.TRANSPARENT);  
 *//activity\_register.xml页面布局文件获取对应的UI控件* btn\_register = (Button)findViewById(R.id.btn\_register);  
 et\_user\_name = (EditText) findViewById(R.id.et\_user\_name);  
 et\_psw =(EditText) findViewById(R.id.et\_psw);  
 et\_psw\_again = (EditText)findViewById(R.id.et\_psw\_again);  
 btn\_register.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 *//获取输入在相应控件中的字符串* getEditString();  
 if (TextUtils.isEmpty(userName)){  
 Toast.makeText(RegisterActivity.this,"请输入用户名",Toast.LENGTH\_LONG).show();  
 }else if (TextUtils.isEmpty(psw)){  
 Toast.makeText(RegisterActivity.this,"请输入密码",Toast.LENGTH\_LONG).show();  
 }else if (TextUtils.isEmpty(pswAgain)){  
 Toast.makeText(RegisterActivity.this,"请再次输入密码",Toast.LENGTH\_LONG).show();  
 }else if (!psw.equals(pswAgain)){  
 Toast.makeText(RegisterActivity.this,"两次密码不一致",Toast.LENGTH\_LONG).show();  
 }else if (isExistUserName(userName)){  
 Toast.makeText(RegisterActivity.this,"此账户名已经存在",Toast.LENGTH\_LONG).show();  
 }else {  
 Toast.makeText(RegisterActivity.this,"注册成功",Toast.LENGTH\_LONG).show();  
 saveRegisterInfo(userName,psw);  
 Intent data = new Intent();  
 data.putExtra("userName",userName);  
 setResult(RESULT\_OK,data);  
 RegisterActivity.this.finish();  
 }  
 }  
 });  
 }  
  
 */\*\*  
 \* 获取控件中的字符串  
 \*/* private void getEditString() {  
 userName = et\_user\_name.getText().toString().trim();  
 psw = et\_psw.getText().toString().trim();  
 pswAgain = et\_psw\_again.getText().toString().trim();  
 }  
  
 */\*\*  
 \* 从SharedPreferences中获取用户名，并判断此用户名是否存在  
 \*/* private boolean isExistUserName(String userName) {  
 boolean has\_userName = false;  
 SharedPreferences sp = getSharedPreferences("loginInfo",MODE\_PRIVATE);  
 String spPwd = sp.getString(userName,"");  
 if (!TextUtils.isEmpty(spPwd)){  
 has\_userName = true;  
 }  
 return has\_userName;  
 }  
  
  
 */\*\*  
 \* 保存用户名和密码到SharedPreferences中  
 \*/* private void saveRegisterInfo(String userName, String psw) {  
 *//把密码用MD5加密* String md5Psw = MD5Utils.md5(psw);  
 *//loginInfo表示文件名* SharedPreferences sp = getSharedPreferences("loginInfo",MODE\_PRIVATE);  
 SharedPreferences.Editor editor = sp.edit(); *//获取编辑器  
 //以用户名为key,密码为value保存到SharedPreferences* editor.putString(userName,md5Psw);  
 editor.apply(); *//提交修改* }  
}

注册按钮按下与弹起有明显的区别，将图片 register\_icon\_normal.png（弹起）、register\_icon\_selected.png（按下）导入 drawable 文件夹中。并在该文件夹中创建背景选择器 register\_selected.xml 文件。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>  
<!-- 注册按钮背景选择器按下为register\_icon\_selected.png，弹起为register\_icon\_normal.png -->*<selector xmlns:android="http://schemas.android.com/apk/res/android">  
 <item android:drawable="@drawable/register\_icon\_selected" android:state\_pressed="true"/>  
 <item android:drawable="@drawable/register\_icon\_normal"/>  
</selector>

注册界面如下。



## 实现MD5加密算法存储账号密码

MD5的全称是Message-Digest ALgorithm 5(信息摘要算法第5版)。把任意长度的字符串变换成固定长度的十六进制字符串。注册与登录都会使用MD5加密算法对密码进行简单的加密。

在utils包中创建一个类MD5Utils。完整代码如下。

package com.hujunyao.cumtorder.utils;  
  
import java.security.MessageDigest;  
import java.security.NoSuchAlgorithmException;  
  
public class MD5Utils {  
 *//创建md5方法对密码进行加密* public static String md5(String text){  
 try {  
 *//获取加密对象digest* MessageDigest digest = MessageDigest.*getInstance*("md5");  
 *//digest方法对密码进行加密* byte[] result = digest.digest(text.getBytes());  
 StringBuilder sb = new StringBuilder();  
 for (byte b : result){  
 int number = b&0xff;  
 String hex = Integer.*toHexString*(number);  
 if (hex.length()==1){  
 sb.append("0"+hex);  
 }else{  
 sb.append(hex);  
 }  
 }  
 return sb.toString();  
 } catch (NoSuchAlgorithmException e) {  
 e.printStackTrace();  
 return "";  
 }  
 }  
}

## 实现登录页面

### 实现登录页面功能

创建一个类，命名为 LoginActivity。在 res/layout 下创建一个 xml 文件，命名为 activity\_login.xml。完整代码如下。

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/login\_bg"  
 android:orientation="vertical">  
 *<!-- 导入标题栏——main\_title\_bar.xml文件 -->* <include layout="@layout/title\_bar" />  
  
 *<!-- 登录界面头像——default\_icon.png -->* <ImageView  
 android:id="@+id/iv\_head"  
 android:layout\_width="70dp"  
 android:layout\_height="70dp"  
 android:layout\_marginTop="25dp"  
 android:layout\_gravity="center\_horizontal"  
 android:src="@mipmap/app\_icon"  
 android:background="@drawable/icon\_bg" />  
  
 *<!-- 用户名输入框 -->* <EditText  
 android:id="@+id/et\_user\_name"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="48dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginLeft="35dp"  
 android:layout\_marginRight="35dp"  
 android:layout\_gravity="center\_horizontal"  
 android:background="@drawable/login\_user\_name\_bg"  
 android:drawableLeft="@drawable/user\_name\_icon"  
 android:drawablePadding="10dp"  
 android:paddingLeft="8dp"  
 android:gravity="center\_vertical"  
 android:hint="请输入用户名"  
 android:singleLine="true"  
 android:textColor="#000000"  
 android:textColorHint="#a3a3a3"  
 android:textSize="14sp" />  
  
 *<!-- 密码输入框 -->* <EditText  
 android:id="@+id/et\_psw"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="48dp"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginRight="35dp"  
 android:layout\_marginLeft="35dp"  
 android:background="@drawable/login\_psw\_bg"  
 android:drawableLeft="@drawable/psw\_icon"  
 android:drawablePadding="10dp"  
 android:paddingLeft="8dp"  
 android:hint="请输入密码"  
 android:inputType="textPassword"  
 android:singleLine="true"  
 android:textColor="#000000"  
 android:textColorHint="#a3a3a3"  
 android:textSize="14sp" />  
  
 *<!-- 登录按钮 -->* <Button  
 android:id="@+id/btn\_login"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="40dp"  
 android:layout\_marginTop="15dp"  
 android:layout\_marginLeft="35dp"  
 android:layout\_marginRight="35dp"  
 android:layout\_gravity="center\_horizontal"  
 android:background="@drawable/register\_selector"  
 android:text="登 录"  
 android:textColor="@android:color/white"  
 android:textSize="18sp" />  
  
 *<!-- 立即注册和找回密码版块 -->* <LinearLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginLeft="35dp"  
 android:layout\_marginRight="35dp"  
 android:layout\_gravity="center\_horizontal"  
 android:orientation="horizontal">  
  
 *<!-- 跳转到注册界面按钮 -->* <TextView  
 android:id="@+id/tv\_register"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:gravity="center\_horizontal"  
 android:padding="8dp"  
 android:text="立即注册"  
 android:textColor="@android:color/white"  
 android:textSize="14sp" />  
  
 *<!-- 跳转到找回密码界面按钮 -->* <TextView  
 android:id="@+id/tv\_find\_psw"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:gravity="center\_horizontal"  
 android:padding="8dp"  
 android:text="找回密码"  
 android:textColor="@android:color/white"  
 android:textSize="14sp" />  
 </LinearLayout>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginLeft="35dp"  
 android:layout\_marginRight="35dp"  
 android:text="账号hjy 密码123"  
 android:gravity="center"  
 android:layout\_gravity="center"  
 android:textColor="@android:color/black"  
 android:background="@color/white"  
 android:textSize="14sp" />  
  
</LinearLayout>

登录界面如下。



### 实现登录页面功能

登录界面主要可以进行登录，跳转到外卖首页，或者点击注册进入账号注册功能。完整代码如下。

package com.hujunyao.cumtorder.activity;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.content.pm.ActivityInfo;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.RelativeLayout;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.hujunyao.cumtorder.R;  
import com.hujunyao.cumtorder.utils.MD5Utils;  
  
  
public class LoginActivity extends AppCompatActivity {  
 private TextView tv\_title;  
 private TextView tv\_back,tv\_register,tv\_find\_psw;  
 private Button btn\_login;  
 private String userName,psw,spPsw;  
 private EditText et\_user\_name,et\_psw;  
 private RelativeLayout rl\_rirle\_bar;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 *//设置页面布局* setContentView(R.layout.activity\_login);  
 *//设置界面为竖屏* setRequestedOrientation(ActivityInfo.SCREEN\_ORIENTATION\_PORTRAIT);  
 init();  
 }  
 */\*\*  
 \* 获取界面控件  
 \*/* private void init() {  
 tv\_title = (TextView)findViewById(R.id.tv\_title);  
 tv\_title.setText("登录");  
 tv\_back = (TextView)findViewById(R.id.tv\_back);  
 tv\_back.setVisibility(View.GONE);  
 rl\_rirle\_bar = (RelativeLayout)findViewById(R.id.title\_bar);  
 rl\_rirle\_bar.setBackgroundColor(Color.TRANSPARENT);  
 tv\_register = (TextView)findViewById(R.id.tv\_register);  
 tv\_find\_psw = (TextView)findViewById(R.id.tv\_find\_psw);  
 btn\_login = (Button) findViewById(R.id.btn\_login);  
 et\_user\_name = (EditText)findViewById(R.id.et\_user\_name);  
 et\_psw = (EditText)findViewById(R.id.et\_psw);  
 *//立即注册控件的点击事件* tv\_register.setOnClickListener(view -> {  
 *//跳转到注册界面* Intent intent = new Intent(LoginActivity.this,RegisterActivity.class);  
 startActivityForResult(intent,1);  
 });  
 *//找回密码控件的点击事件* tv\_find\_psw.setOnClickListener(v ->  
 Toast.makeText(LoginActivity.this, "联系管理员吧，这儿弄不了", Toast.LENGTH\_SHORT).show()  
 );  
 *//登录按钮的点击事件* btn\_login.setOnClickListener(view -> {  
 userName = et\_user\_name.getText().toString().trim();  
 psw = et\_psw.getText().toString().trim();  
 String md5Psw = MD5Utils.md5(psw);  
 spPsw = readPsw(userName);  
 *//如果用户名为空* if (TextUtils.isEmpty(userName)){  
 Toast.makeText(LoginActivity.this,"请输入用户名",Toast.LENGTH\_LONG).show();  
 }else if (TextUtils.isEmpty(psw)){ *//如果密码为空* Toast.makeText(LoginActivity.this,"请输入密码",Toast.LENGTH\_LONG).show();  
 }else if (md5Psw.equals(spPsw)){  
 Toast.makeText(LoginActivity.this,"登录成功",Toast.LENGTH\_LONG).show();  
 *//保存登录状态和登录的用户名* saveLoginStatus(true,userName);  
 *//把登录成功的状态传递到MainActivity中* Intent intent = new Intent(LoginActivity.this, ShopActivity.class);  
 LoginActivity.this.finish();  
 startActivity(intent);  
  
 }else if (!TextUtils.isEmpty(spPsw) && !md5Psw.equals(spPsw)){  
 Toast.makeText(LoginActivity.this,"输入的用户名和密码不一致",Toast.LENGTH\_LONG).show();  
 }else {  
 Toast.makeText(LoginActivity.this,"此用户名不存在",Toast.LENGTH\_LONG).show();  
 }  
 });  
 }  
 */\*  
 \* 从SharedPreferences根据用户名获取密码  
 \*/* private String readPsw(String userName){  
 SharedPreferences sp = getSharedPreferences("loginInfo",MODE\_PRIVATE);  
 return sp.getString(userName,"");  
 }  
 */\*  
 \* 保存登录状态和登录用户名到SharedPreferences中  
 \*/* private void saveLoginStatus(boolean status, String userName) {  
 SharedPreferences sp = getSharedPreferences("loginInfo",MODE\_PRIVATE);  
 SharedPreferences.Editor editor = sp.edit(); *//获取编辑器* editor.putBoolean("isLogin",status);  
 editor.putString("loginUserName",userName); *//保存登录的用户名* editor.commit(); *//提交修改* }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (data!=null){  
 *//从注册界面传递过来的用户名* String userName = data.getStringExtra("userName");  
 if (!TextUtils.isEmpty(userName)){  
 et\_user\_name.setText(userName);  
 *//设置光标的位置* et\_user\_name.setSelection(userName.length());  
 }  
 }  
 }  
}

# 遇到的问题与解决方法

## 打开 AVD 时一直黑屏

解决办法：

1. 切换版本较低的 AVD；
2. 降低 AVD 内存；
3. 在 SDK Manager 中安装适用该系统的附加 SDK Tools，如 Intel x86 系统安装 Intel x86 Emulator Accelerator (HAXM installer)。

## HAXM 安装错误

解决办法：

1. 计算机管理中，把VT服务关闭；
2. BIOS中的VT-x相关的改为enable；
3. 如果是重装，需要注意完全卸载之前的版本。

## 测试类报错 Cannot resolve symbol 'Test'

解决办法：

1. 删除 build.gradle 中所有的测试相关的依赖testImplementation；
2. 加入 testImplementation 'junit:junit:4.12'。

## build 中文乱码问题

解决办法：

在 Android Studio 中双击 Shift 或者点击顶部全局搜索 Edit Custom VM Options，在打开的文件中添加一句 -Dfile.encoding=UTF-8，再重启Android Studio即可。

## okhttp3 3.14版本导致app闪退

解决办法：

okhttp 还需要 okio 包依赖，所以要下载两个都要导入，但是尝试下载各种版本后都不能运行，最后直接导入网络依赖成功，即改为 implementation 'com.squareup.okhttp3:okhttp:5.0.0-alpha.3' 即可成功，网络依赖导入会自动链接 okio。

## Tomcat 闪退

解决办法：

因为之前只有一个 JDK 版本，自己修改后装了两个 jdk 版本，tomcat 没有直接链接到设置的环境变量，因此要手动设置一下 tomcat 的参数，在【Apache Tomcat 9.0 Tomcat9 Properties -> Java -> Java Virtual Machine】修改为如正确的 JDK路径，如：

D:\Software\JavaJDK\jdk1.8.0\_311\jre\bin\server\jvm.dll。

## 报错 Resources$NotFoundException: String resource ID xxxx

解决办法：

Toast.makeText()，textView.setText 之类的函数通常有几个重载，如： textView.setText(CharSequence text); textView.setText(int resId); 如果不小心将一个 int 值传给了它，那它不会显示该 int 值，而是跑到工程下去找一个对应的 resource 的 id，当然是找不到的，于是就报错。

如果要显示该 int 值，就要将 int 转化成 String，可以在该 int 值后面+""，强制转为 String。需要将 ShopAdapter.java 中的vh.xxx.setText 中仅存为数字的改为String类型，或者直接写成一个完整的句子。

## 设置了 background 后，button 依旧是默认颜色

解决办法：

把themes.xml 这个文件 Style 的 parent 这里修改为如下：

Theme.MaterialComponents.DayNight.NoActionBar.Bridge

## 华为、荣耀系列手机开始 ADB 调试后也不能连接到 Android Studio

解决办法：

还需要在手机设置里开启允许 HDB 调试才行。

## 连接不到 tomcat，就连 OnFailure 函数都不进入

解决办法：

在 AndroidManifest.xml 中加入网络权限。

name="android.permission.INTERNET"

name="android.permission.ACCESS\_NETWORK\_STATE"

android:usesCleartextTraffic="true"

# 参考文献

1. android项目实战-订餐项目. 码龙老师. bilibili. https://www.bilibili.com/video/BV1Hv411z79k
2. Android订餐项目期末大作业. 喜欢熊猫的卡卡. bilibili. https://www.bilibili.com/video/BV1rf4y1C7cn
3. Banner 2.0. youth5201314. Github. https://github.com/youth5201314/banner

# 实验感悟

Android Studio 是一个非常有用的软件，通过本次学习，了解了如何制作好一款软件，好的软件更需要优化的内层代码逻辑和外表的美观界面，两者都具备是非常不容易的。这次实验主要是跟着网上的教程进行了学习与复现，自学的能力真的很重要，同时也要注重进行一些创新。完成本次设计的过程比较艰难，也体现自己的能力不足，需要在今后加强学习，主动研究，从而加强自己的不可替代性，而不是一直造低质量的轮子。