

# 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
1.1 课设目的.....	1
1.2 课设要求.....	1
2 课设概述.....	1
3 开发环境配置.....	1
3.1 操作系统.....	1
3.2 开发工具.....	1
3.3 API 版本 .....	1
3.4 主要的库和依赖版本.....	1
4 所需环境与库安装准备过程.....	2
5 功能实现.....	6
5.1 店铺功能业务实现.....	6
5.1.1 搭建标题栏布局.....	6
5.1.2 搭建广告栏布局（轮播广告） .....	8
5.1.3 搭建店铺页面布局.....	10
5.1.4 搭建店铺列表条目界面布局.....	12
5.1.5 封装店铺信息与菜品信息的实体类.....	14
5.1.6 编写广告栏的适配器.....	19
5.1.7 编写店铺列表适配器.....	20
5.1.8 实现店铺界面显示功能.....	22
5.2 店铺功能业务实现.....	24
5.2.1 实现店铺界面显示功能.....	24
5.2.2 搭建菜单列表条目界面布局.....	31
5.2.3 搭建购物车列表条目界面布局.....	34
5.2.4 搭建确认清空购物车界面布局.....	35
5.2.5 编写店铺列表适配器.....	37
5.2.6 编写店铺列表适配器.....	39
5.2.7 编写店铺列表适配器.....	42
5.3 菜品详情功能业务实现.....	50
5.3.1 搭建菜品详情界面布局.....	50

5.3.2 实现菜品界面显示功能 .....	51
5.4 订单功能业务实现 .....	52
5.4.1 搭建订单界面布局 .....	53
5.4.2 搭建订单列表条目界面布局 .....	57
5.4.3 搭建支付界面布局 .....	59
5.4.4 编写订单列表适配器 .....	59
5.4.5 实现订单显示与支付功能 .....	62
5.5 实现欢迎页面 .....	64
5.5.1 实现欢迎页面布局 .....	64
5.5.2 实现欢迎页面功能 .....	65
5.6 实现注册页面 .....	67
5.6.1 实现注册页面功能 .....	67
5.6.2 实现注册页面功能 .....	69
5.7 实现 MD5 加密算法存储账号密码 .....	73
5.8 实现登录页面 .....	73
5.8.1 实现登录页面功能 .....	73
5.8.2 实现登录页面功能 .....	77
6 遇到的问题与解决方法 .....	80
6.1 打开 AVD 时一直黑屏 .....	80
6.2 HAXM 安装错误 .....	80
6.3 测试类报错 Cannot resolve symbol 'Test' .....	80
6.4 build 中文乱码问题 .....	80
6.5 okhttp3 3.14 版本导致 app 闪退 .....	80
6.6 Tomcat 闪退 .....	80
6.7 报错 Resources\$NotFoundException: String resource ID xxxx .....	81
6.8 设置了 background 后, button 依旧是默认颜色 .....	81
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
6.10 连接不到 tomcat, 就连 onFailure 函数都不进入 .....	81
7 参考文献 .....	81
8 实验感悟 .....	81

---

# 1 课设目的与要求

## 1.1 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。

## 1.2 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。

## 2 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。

## 3 开发环境配置

### 3.1 操作系统

Windows 10 专业版 20H2

### 3.2 开发工具

Android Studio Arctic Fox | 2020.3.1 Patch 4  
Tomcat 9.0

### 3.3 API 版本

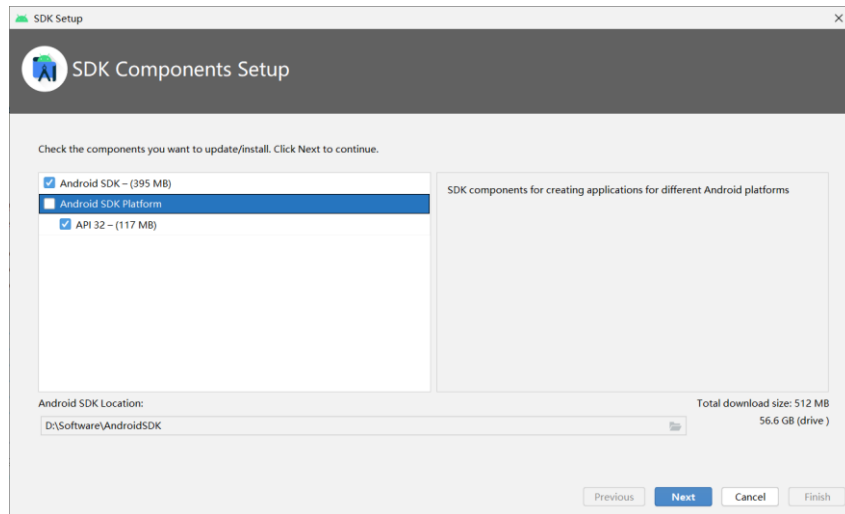
API 选择 30 (R Android 11.0)


### 3.4 主要的库和依赖版本

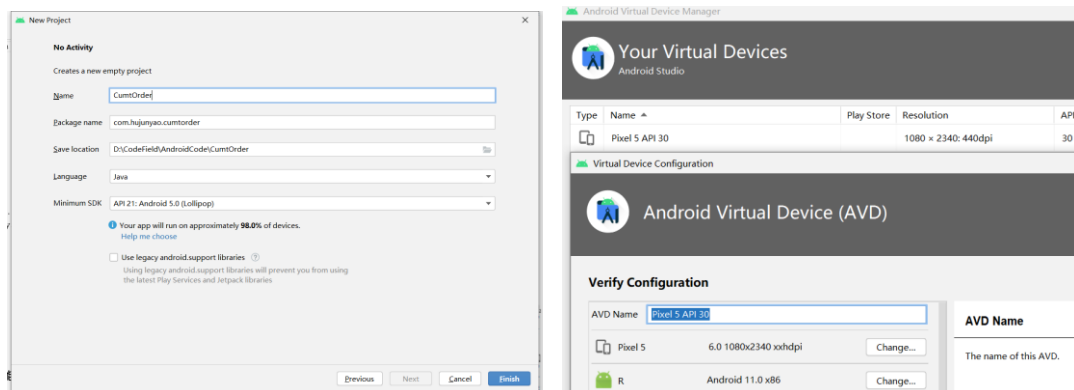
glide 3.7.0  
gson 2.8.5  
okhttp:5.0.0-alpha.3

## 4 所需环境与库安装准备过程

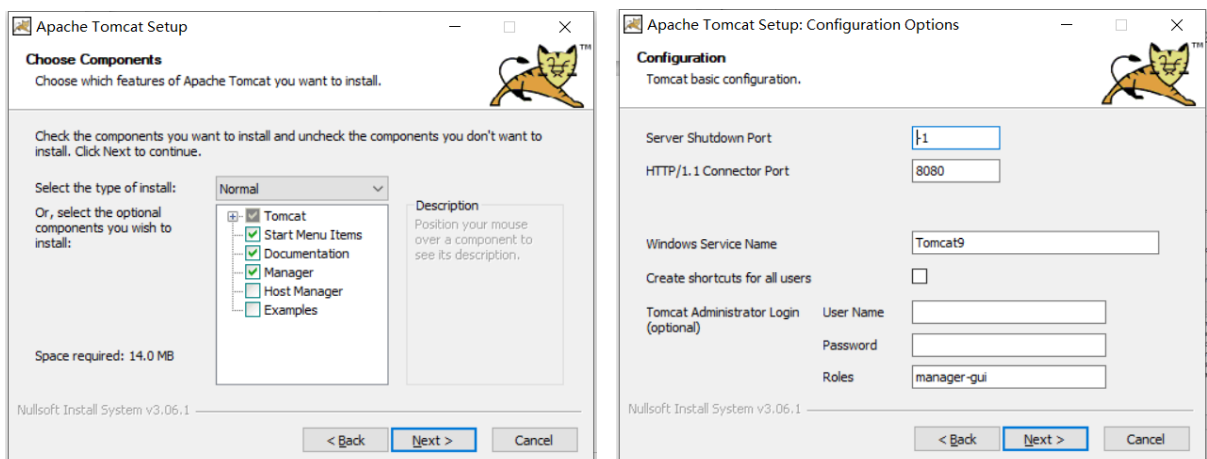
安装 SDK，选择合适的目录。



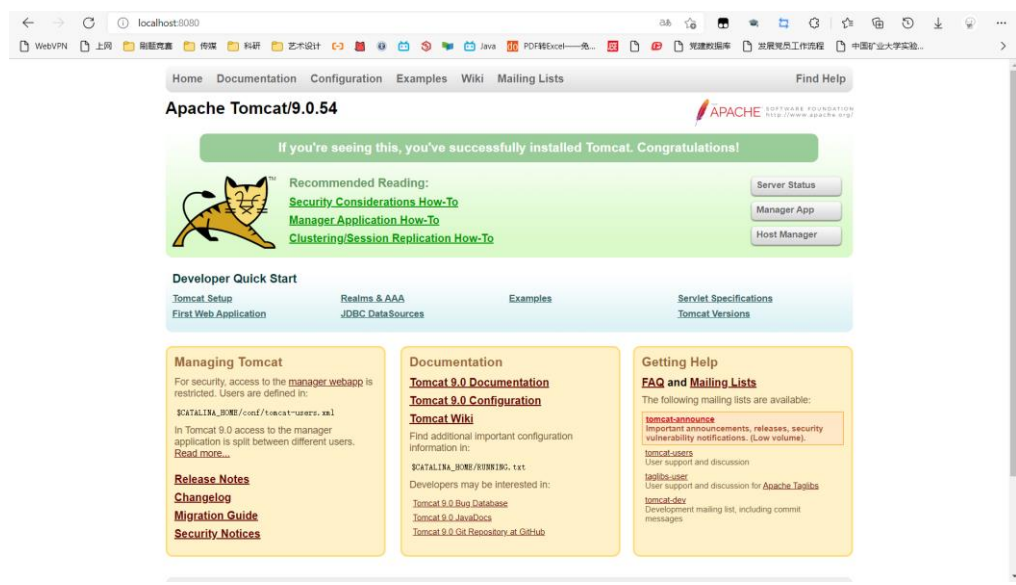
在 Android Studio 中选择【File -> New -> New Project】，选取 No Activity，创建名称为 CumtOrder 的 Java 工程，其他参数如下。点击  可选择虚拟设备，选择 Pixel 5 作为虚拟机，API 选择 30。



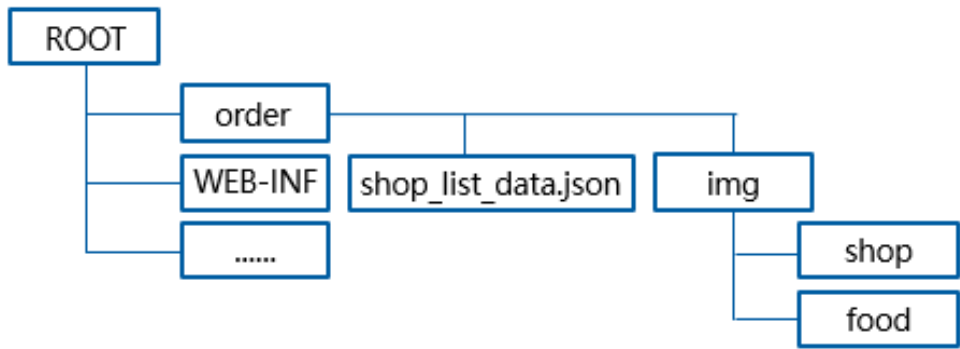
在 Tomcat 官网下载 Tomcat 9.0，进行安装操作，选择默认即可。



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。



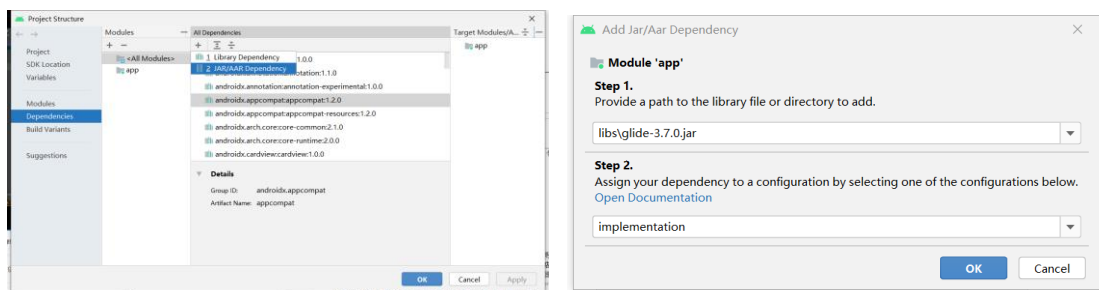
```
[
{
  "id":1,
  "shopName":"思维特精品小碗菜",
  "saleNum":8544,
  "offerPrice":10,
  "distributionCost":0,
  "welfare":"首单减 6",
  "time":50,
  "shopPic":"/img/shop/shop1.png",
  "shopNotice":"徐州小碗菜热销榜第一名",
  "foodList":[
  {
```

```

"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。

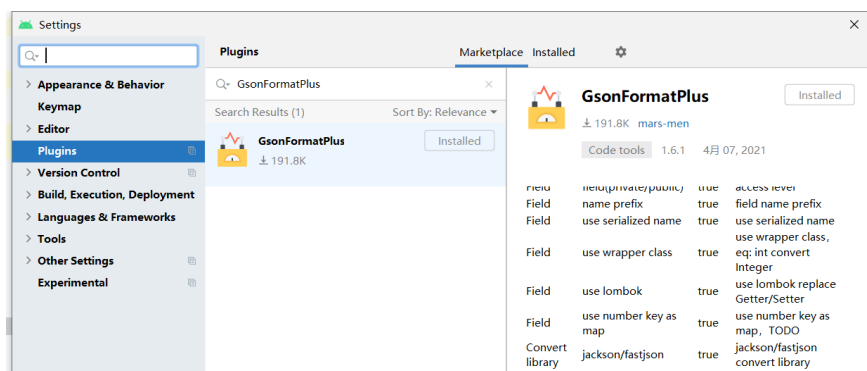
```

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'
}

```



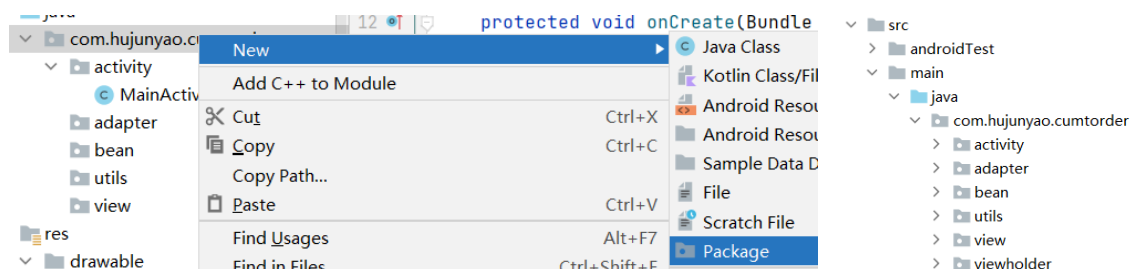
安装插件的过程如下，以安装 GsonFormatPlus 为例，打开【File -> Settings -> Plugins】，搜索 GsonFormatPlus，直接安装即可。



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.



## 5 功能实现

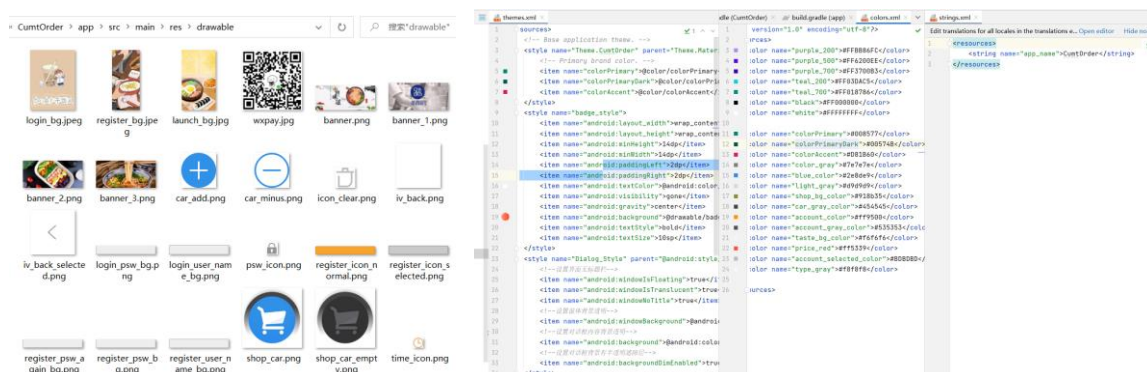
### 5.1 店铺功能业务实现

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。

#### 5.1.1 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"

```

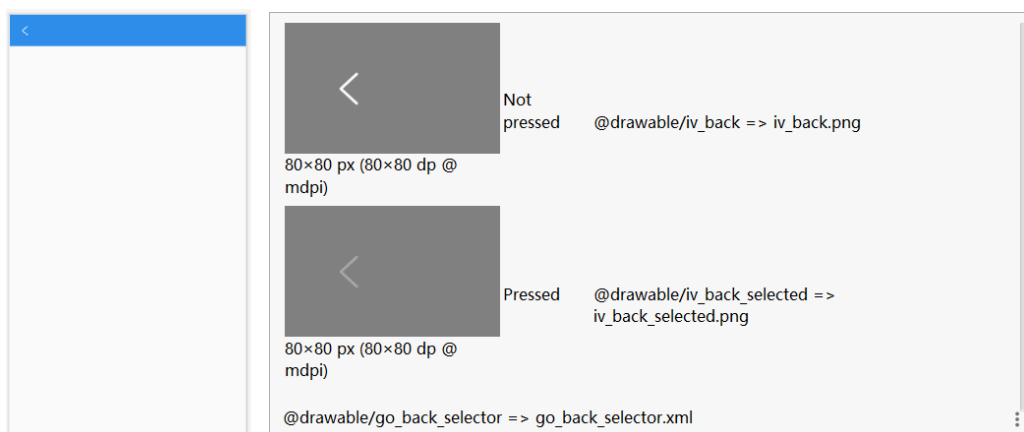
在 themes.xml 中去掉默认标题（一定要加 .Bridge）

```

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

```

当前效果如下。



### 5.1.2 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>

```

### 5.1.3 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>

```

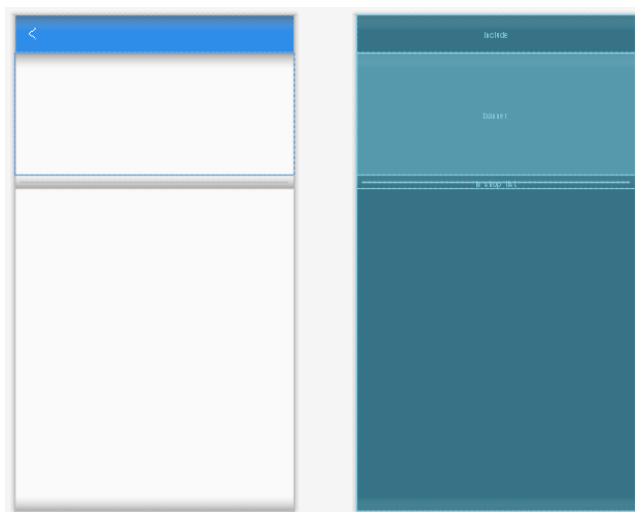
在 view 包创建 ShopListView 类继承 ListView 类。完整代码如下。

```
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。



---

### 5.1.4 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>
</RelativeLayout>
```



```

<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>
```

### 5.1.5 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;
    }
}

```

---

### 5.1.6 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);
    }

}
```

### 5.1.7 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;
}
}

```

### 5.1.8 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").bui
ld();

    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) { }

```

## 5.2 店铺功能业务实现

### 5.2.1 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>

```

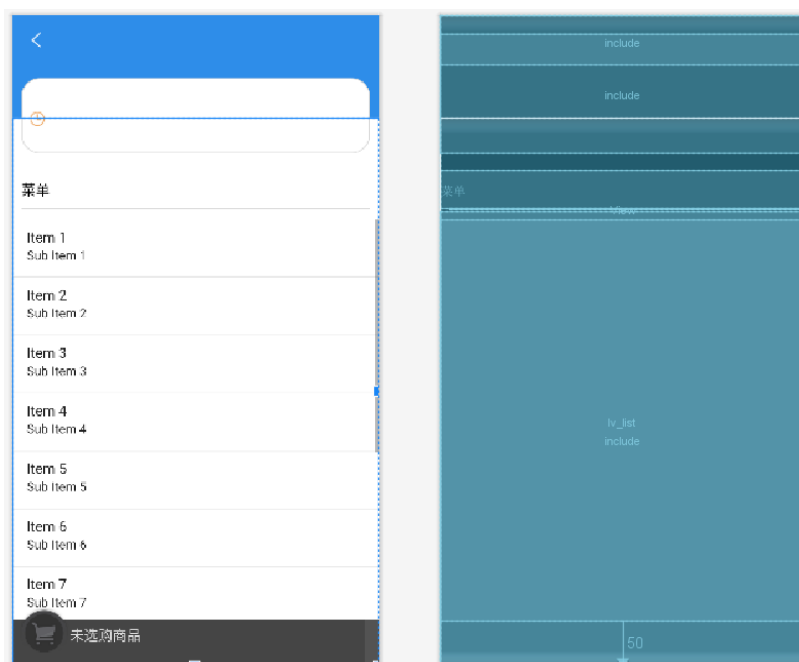
创建 corner\_bg.xml 文件。完整代码如下。

```
<?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。



## 5.2.2 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。

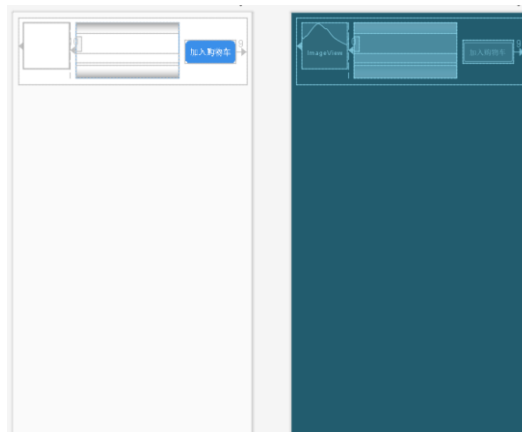
```

<?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。



### 5.2.3 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" />

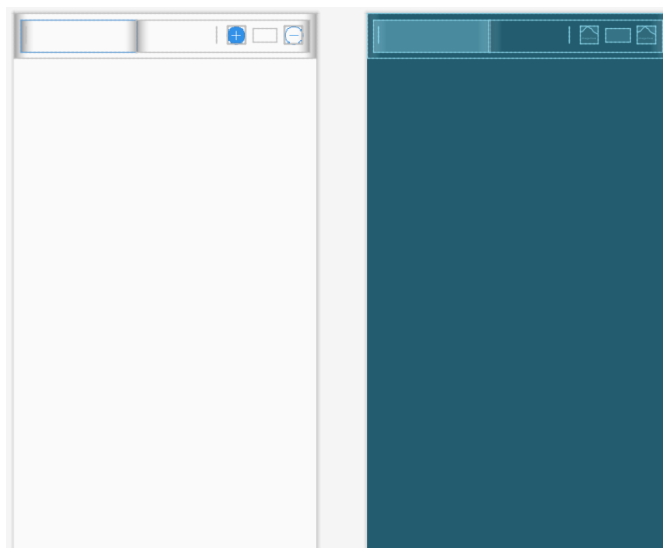
<TextView
    android:id="@+id/tv_food_count"
    android:layout_width="34dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="10dp"
    android:layout_marginEnd="10dp"
    android:gravity="center"
    android:textColor="@android:color/black" />

<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。



#### 5.2.4 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。

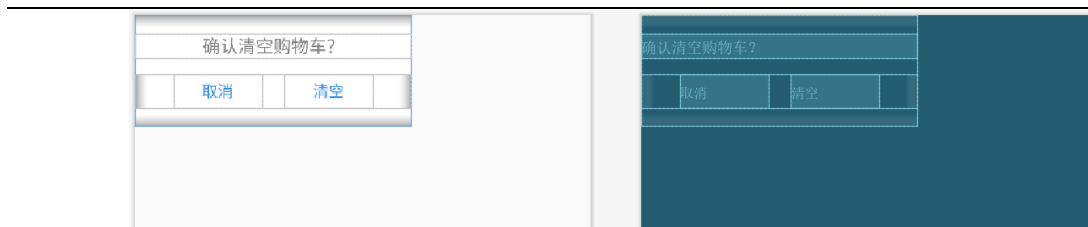
```

<?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。





### 5.2.5 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
}
}

```

## 5.2.6 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);
    }
}

```

## 5.2.7 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
    }
    @SuppressWarnings("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()
{
            @SuppressWarnings("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
                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

    }
    @SuppressWarnings("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
}

@SuppressWarnings("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
        });
    }

    @SuppressWarnings("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
}

@SuppressWarnings("HandlerLeak")
class MHandler extends Handler {
    @SuppressWarnings("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);
            }
        }
    }
}

```



## 5.3 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。

### 5.3.1 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。

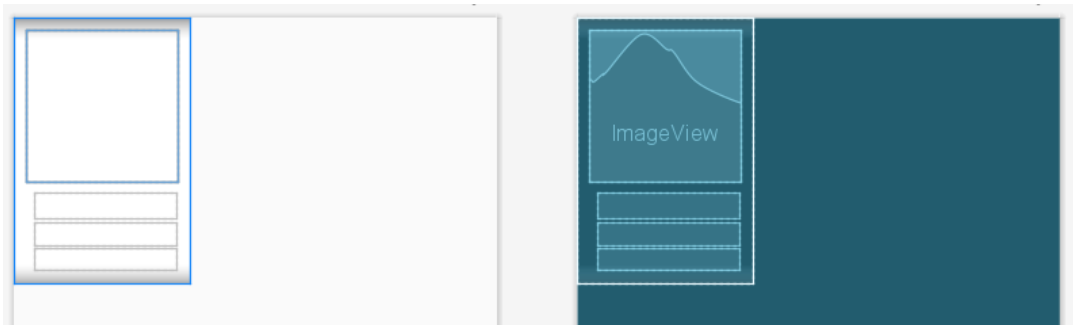
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:background="@android:color/white"
    android:padding="10dp"
    android:orientation="vertical">
    <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。



### 5.3.2 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);
    }
}

```

## 5.4 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。



### 5.4.1 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>

```

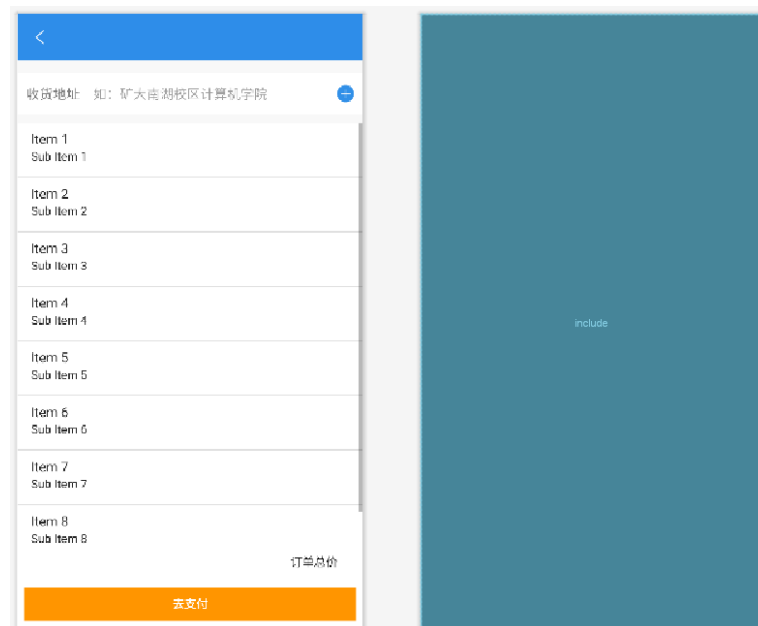
创建背景选择器 `payment_bg_selector.xml`，完整代码如下。

```

<?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。



#### 5.4.2 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。

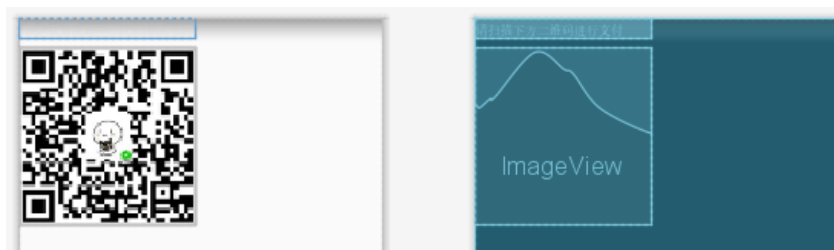


### 5.4.3 搭建支付界面布局

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。



### 5.4.4 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;
    }
}

```

---

### 5.4.5 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)));
}
}

```

## 5.5 实现欢迎页面

### 5.5.1 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" />

```

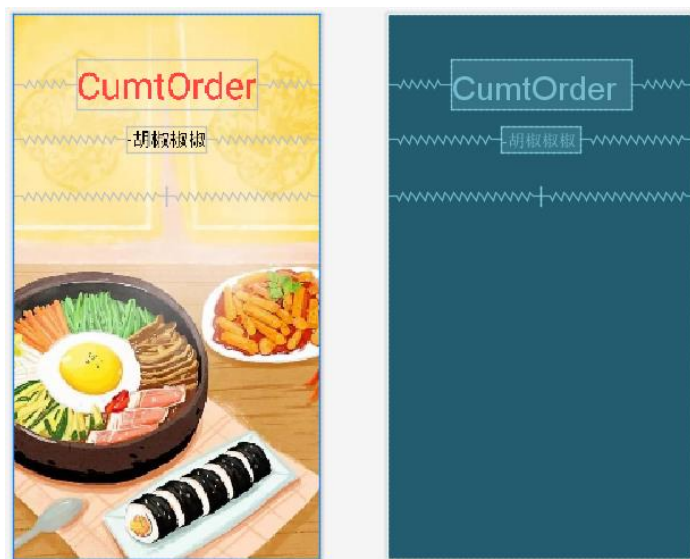
```

<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。



### 5.5.2 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
            }
        };
        timer.schedule(task, 3000);
    }
}

```

```

        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
}
}

```

## 5.6 实现注册页面

### 5.6.1 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>

```

### 5.6.2 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
}
}

```

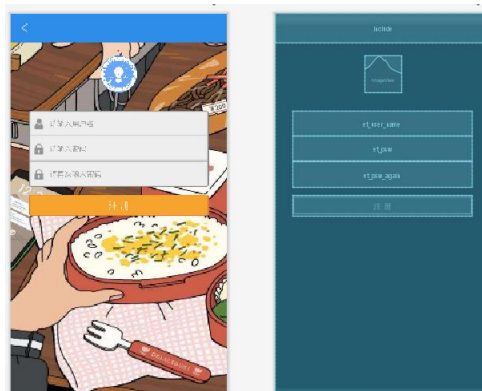
注册按钮按下与弹起有明显的区别，将图片 `register_icon_normal.png`（弹起）、`register_icon_selected.png`（按下）导入 `drawable` 文件夹中。并在该文件夹中创建背景选择器 `register_selected.xml` 文件。完整代码如下。

```

<?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。



## 5.7 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 "";
        }
    }
}
```

## 5.8 实现登录页面

### 5.8.1 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。





---

### 5.8.2 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());
        }
    }
}
}
}
}

```

---

## 6 遇到的问题与解决方法

### 6.1 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)。

### 6.2 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。

### 6.3 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'。

### 6.4 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。

### 6.5 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。

### 6.6 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。

---

## 6.7 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` 值后面加 `""`, 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。

## 6.8 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`

## 6.9 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。

## 6.10 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"`

## 7 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>

## 8 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, 同时也要注意进行一些创新。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。