

**寻 味 APP**

**需**

**求**

**分**

**析**

**说**

**明**

**书**

**小组: 第七组**

**成员: 游贤平、伏俊奇、黄浩琛、**

**向旭华、罗漫、伏丽云**

目录

[1、绪论 2](#_Toc518594244)

[1.1 、课题研究的背景及意义 2](#_Toc518594245)

[1.2 、项目介绍 3](#_Toc518594246)

[.1.3、开发工具及运行环境 3](#_Toc518594247)

[2、项目分析 3](#_Toc518594248)

[2.1、需求分析 3](#_Toc518594249)

[2.2、可行性分析 4](#_Toc518594250)

[3、项目设计 4](#_Toc518594251)

[3.1，设计目标 4](#_Toc518594252)

[3.2，数据库慨念设计结构 4](#_Toc518594253)

[3.3、功能结构图 6](#_Toc518594254)

[4、主要功能模块实现 7](#_Toc518594255)

[4.1、登陆界面 7](#_Toc518594256)

[4.2、注册界面 8](#_Toc518594257)

[4.3、主页界面 9](#_Toc518594258)

[4.4、订单界面 9](#_Toc518594259)

[4.5、我的界面 10](#_Toc518594260)

[4.6、引导页界面 10](#_Toc518594261)

[5、总结 11](#_Toc518594262)

[附录1 数据库设计代码 11](#_Toc518594263)

[附录2 Android 端部分代码 13](#_Toc518594264)

# 1、绪论

## 、课题研究的背景及意义

随着社会的发展，经济水平的提高，手机已成为人们如影随形的伙伴，同时伴随移动互联网的更为深入的发展，手机的功能也变得越来越丰富，。手机订餐App也随之融入到了人们的生活中去，成了我们生活中的必不可少的工具，但是，订餐类的App屡见不鲜，如：element，美团，大众点评，百度外卖等等，它们功能如出一辙，没有突出特点，市场目前需要的是创新，且根据市场分析目前网上订餐的主空军为中年及以下，市场深广，用户量广，根据此特性，专门定制了一份属于他们的点餐计划，将为这些人开启“寻味”之旅！

## 、项目介绍

此订餐手机App主要采用的是Android Studio作为App架构设计以及MyEclipse作为后台数据处理，以操作便捷，界面友好，灵活，适用为出发点，采用了模块化设计方法，App实现了以下基本功能。

**APP端：**

用户登陆：可以实现用户账号的登陆，判断当前用户的状态；

用户注册：实现了用户信息的注册；

商品：商品购买；

订单：增删改查；

**Web服务端：**

管理员登陆

商品：增删改查

**特点：**

方便性：方便用户及时了解

灵活性：App性能稳定

实用性：由于App页面简介，很容易操作

## .1.3、开发工具及运行环境

开发环境：Android studio

服务器环境：Tomcat 7.0+ myeclipse mysql

APP运行环境： Android版本5.1

# 2、项目分析

## 2.1、需求分析

通过调查，要求App具备一下功能

A 用户登陆

B 用户注册

C商品的搜索

D订单的增删改查

E个人信息修改

F退出登陆

## 2.2、可行性分析

手机App使餐饮业走上了科学化，正规化的道路，

经济可行性：适用手机订餐对消费者来说可以节约大量的时间，对于餐饮业来说可以减少宣传成本，减少人力成本

技术可行性：目前国内外软硬件产品和技术迅速发展，服务器操作系统等资源容易获得，并有较大的选择余地，该APP要求适中，在现有的技术条件完全可以满足

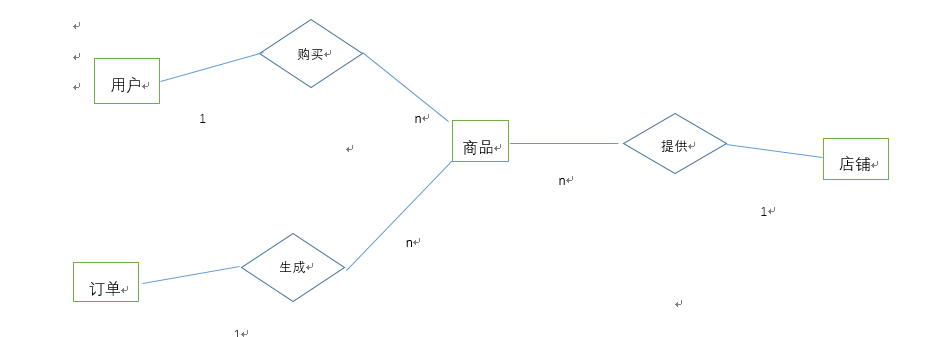
# 3、项目设计

## 3.1，设计目标

该APP主要是针对用户订餐过程中设计的主要实现以下目标：合法用户可以登陆该APP，新用户可以注册，商品搜索，对订单的增删改查

## 3.2，数据库慨念设计结构

E-R图



根据上述E-R模型，将之转换为关系模式集：

关系1：用户（用户序列号，用户名，用户密码，用户真实姓名，用户性别，用户手机号，用户地址，用户余额，生日，头像）；

关系2：店铺（店铺序列号，店铺编号，店铺名，注册人账号，店铺简介，店铺地址，店铺电话，营业时间）；

关系3：美食（商品序列号，商品编号，店铺编号，商品名称, 商品简介, 商品单价，商品图片）；

关系4：订单（订单编号，用户账号，店铺编号，用户姓名，订单数量，总价）；

数据库信息表

用户表(t\_user)

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 属性 | 长度 | 备注 |
| U\_id | Int |  | 用户序列号 |
| U\_account | varcher | 10 | 用户名，主键 |
| U\_password | varchar | 18 | 用户密码 |
| U\_name | varchar | 10 | 用户真实姓名 |
| U\_sex | varchar | 3 | 用户性别 |
| U\_phone | varchar | 12 | 用户手机号 |
| U\_address | varchar | 50 | 用户地址 |
| U\_balance | varcher | 10 | 用户余额 |
| U\_birthday | varcher | 20 | 生日 |
| U\_headP | varcher | 9 | 头像 |

商品表(t\_foods)

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 属性 | 长度 | 备注 |
| F\_id | int |  | 商品序列号 |
| F\_number | varcher | 11 | 商品编号 |
| F\_name | Varcher | 11 | 商品名称 |
| F\_price | decimal | 225 | 商品价格 |
| S\_number | varcher | 10 | 店铺ID |
| F\_introduce | Varcher | 225 | 商品介绍 |
| F\_images | varcher | 225 | 商品图片 |

订单表(t\_order)

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 属性 | 长度 | 备注 |
| O\_id | int |  | 订单ID |
| O\_number | varcher | 11 | 订单编号 |
| U\_account | Varcher | 225 | 用户账号 |
| S\_number | varcher | 11 | 店铺编号 |
| O\_num | Varcher | 11 | 订单数量 |
| O\_price | decimal | 10 | 订单价格 |
| O\_total | decimal | 10 | 订单总数 |

店铺表 (t\_shop)

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 属性 | 长度 | 备注 |
| S\_id | int |  | 店铺ID |
| S\_nember | varcher | 255 | 店铺编号 |
| U\_account | Varcher | 255 | 注册人账号 |
| S\_introduction | Varcher | 255 | 店铺介绍 |
| S\_address | Varcher | 255 | 地址 |
| S\_phone | Varcher | 255 | 电话 |
| S\_heap | Varcher | 255 | 头像 |
| S\_tuijian | Varcher | 255 | 推荐 |
| S\_worktime | varcher | 255 | 营业时间 |

## C:\Users\X\Desktop\FindFoods.jpg3.3、功能结构图

# 4、主要功能模块实现

## 4.1、登陆界面

进入App系统会自动判别当前登陆状态，然后提示登陆，系统会查询User，如果用户存在且正确，用户则进行相关操作。如果用户不存在则显示密码错误或提示注册

## 4.2、注册界面

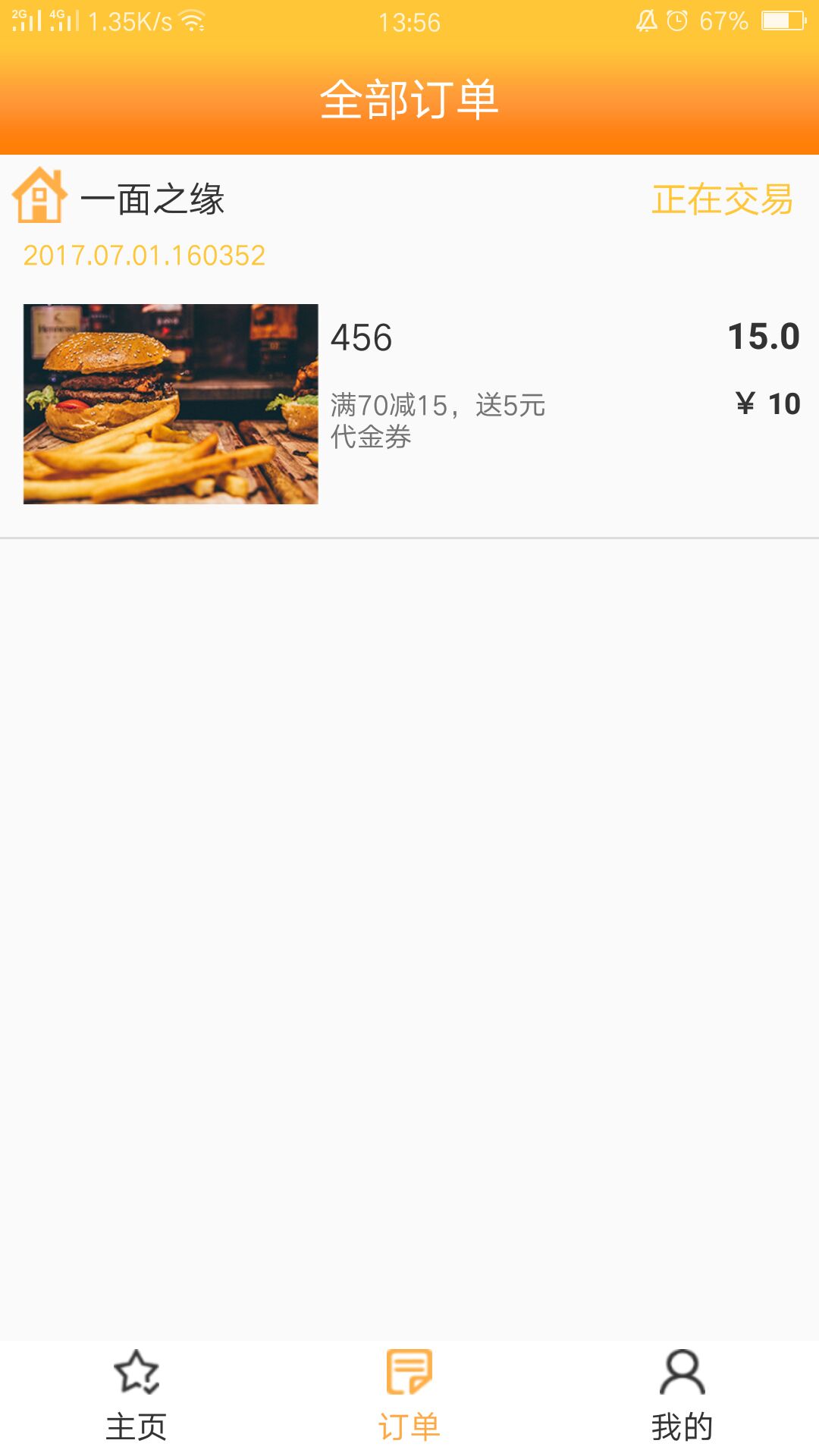


## 4.3、主页界面





## 4.4、订单界面



## 4.5、我的界面



## 4.6、引导页界面



# 5、总结

经过紧张而又刺激的两个月的期末项目准备，我们最终迎来了答辩。

5月4号开始到7月6号，为期两个月的时间完成了我们的项目，完成了我的第一个完整的移动端项目，这个项目为期63天的时间，从切图到交互、从个人到团队合作每个阶段都有不一样的发现，以及不一样的成长和收获。

项目伊始，我们组遇到最大难题是移动端与服务端的交互，因为无例可鉴，之前都是用自带的sqllite，所以对这个还是比较陌生的，很茫然，通过查询大量的例子，做这个差不多花费了一个多月时间，即便如此这也难不倒我们，，最终还是做出来了。UI的界面设计也很是艰难，通过查找资料，借鉴类似APP，例如美团、百度外卖等，通过对比分析最终设计呈现出我们现在的App界面

在今天项目结束的时候感觉自己在移动端的感触颇多，自接下来中再遇到移动端的就轻松多了！一步一个脚印，在做好了自己的当前工作后，也要增强自己的技能，不断的提升不断的打破！

# 附录1 数据库设计代码

/\*

Navicat MySQL Data Transfer

Source Server : root

Source Server Version : 50717

Source Host : localhost:3306

Source Database : xw\_sql

Target Server Type : MYSQL

Target Server Version : 50717

File Encoding : 65001

Date: 2018-07-05 12:33:23

\*/

SET FOREIGN\_KEY\_CHECKS=0;

-- ----------------------------

-- Table structure for tb\_foods

-- ----------------------------

DROP TABLE IF EXISTS `tb\_foods`;

CREATE TABLE `tb\_foods` (

`F\_id` int(11) NOT NULL AUTO\_INCREMENT,

`F\_number` varchar(11) NOT NULL,

`S\_name` varchar(100) NOT NULL,

`F\_name` varchar(100) NOT NULL,

`F\_price` decimal(10,2) NOT NULL,

`F\_introduce` varchar(250) NOT NULL,

`F\_images` varchar(100) NOT NULL,

PRIMARY KEY (`F\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=19303 DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of tb\_foods

-- ----------------------------

INSERT INTO `tb\_foods` VALUES ('1', '201', '一面之缘', '红灯照牛肉面', '15.00', '非常好吃，非常好吃，非常好吃，非常好吃，非常好吃', 'c1');

INSERT INTO `tb\_foods` VALUES ('2', '202', 'Mr.龙虾先生（新南店）', '小龙虾', '58.00', '非常好吃，非常好吃，非常好吃，非常好吃，非常好吃', 'c2');

INSERT INTO `tb\_foods` VALUES ('3', '203', '锦城印象火锅酒店（高新店）', '火锅', '107.00', '非常好吃，非常好吃，非常好吃，非常好吃，非常好吃', 'c3');

INSERT INTO `tb\_foods` VALUES ('4', '204', '万州烤鱼', '万州烤鱼', '63.00', '非常好吃，非常好吃，非常好吃，非常好吃，非常好吃', 'c4');

INSERT INTO `tb\_foods` VALUES ('5', '205', '大蓉和（拉德芳斯精品店）', '鱼香茄子', '12.00', '非常好吃，非常好吃，非常好吃，非常好吃，非常好吃', 'c5');

-- ----------------------------

-- Table structure for tb\_older

-- ----------------------------

DROP TABLE IF EXISTS `tb\_older`;

CREATE TABLE `tb\_older` (

`O\_id` int(11) NOT NULL,

`O\_number` varchar(50) NOT NULL,

`U\_account` varchar(20) NOT NULL,

`O\_num` int(11) NOT NULL,

`O\_price` decimal(10,2) NOT NULL,

`O\_total` decimal(10,2) NOT NULL,

`S\_name` varchar(100) NOT NULL,

`O\_state` varchar(20) NOT NULL,

`F\_name` varchar(255) DEFAULT NULL,

PRIMARY KEY (`O\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of tb\_older

-- ----------------------------

INSERT INTO `tb\_older` VALUES ('1', '2017.07.01.160352', '123567', '2', '30.00', '15.00', '一面之缘', '正在交易', '456');

INSERT INTO `tb\_older` VALUES ('2', '2017.6.29.123055', '585697', '1', '63.00', '63.00', '万州烤鱼', '正在交易', '456');

INSERT INTO `tb\_older` VALUES ('3', '20180530130036', '612588', '1', '58.00', '58.00', 'Mr.龙虾先生（新南店）', '正在交易', '456');

-- ----------------------------

-- Table structure for tb\_shops

-- ----------------------------

DROP TABLE IF EXISTS `tb\_shops`;

CREATE TABLE `tb\_shops` (

`S\_id` int(11) NOT NULL AUTO\_INCREMENT,

`S\_number` varchar(20) NOT NULL,

`U\_account` varchar(20) NOT NULL,

`S\_introduction` varchar(250) NOT NULL,

`S\_address` varchar(100) NOT NULL,

`S\_phone` varchar(12) NOT NULL,

`S\_headP` varchar(20) NOT NULL,

`S\_tuiJianP` varchar(20) NOT NULL,

`S\_workTime` varchar(5) NOT NULL,

`S\_type` varchar(30) NOT NULL,

`S\_name` varchar(100) NOT NULL,

`S\_price` decimal(10,2) NOT NULL,

PRIMARY KEY (`S\_id`,`S\_number`)

) ENGINE=InnoDB AUTO\_INCREMENT=16306 DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of tb\_shops

-- ----------------------------

INSERT INTO `tb\_shops` VALUES ('16301', '1', '812567', '川菜，中餐', '高新区锦城大道', '13892856781', 'z1', 'b1', '15', '4', '大蓉和（拉德芳斯精品店）', '87.00');

INSERT INTO `tb\_shops` VALUES ('16302', '2', '987568', '面食，抄手', '高新区苏宁广场', '13755612857', 'z2', 'b2', '12', '4', '一面之缘', '15.00');

INSERT INTO `tb\_shops` VALUES ('16303', '3', '555667', '小龙虾，湘菜', '火车南站，凯德广场', '13688621357', 'z3', 'b3', '15', '1', 'Mr.龙虾先生（新南店）', '58.00');

INSERT INTO `tb\_shops` VALUES ('16304', '4', '666338', '火锅，中餐', '成都南站', '13955321758', 'z4', 'b4', '12', '1', '锦城印象火锅酒店（高新店）', '107.00');

INSERT INTO `tb\_shops` VALUES ('16305', '5', '925681', '川菜，烤鱼', '华西医院，省体育馆', '13566289213', 'z5', 'b5', '12', '1', '万州烤鱼', '63.00');

-- ----------------------------

-- Table structure for tb\_user

-- ----------------------------

DROP TABLE IF EXISTS `tb\_user`;

CREATE TABLE `tb\_user` (

`U\_id` int(11) NOT NULL AUTO\_INCREMENT,

`U\_account` varchar(20) NOT NULL,

`U\_password` varchar(20) NOT NULL,

`U\_name` varchar(25) DEFAULT NULL,

`U\_sex` varchar(4) DEFAULT NULL,

`U\_phone` varchar(12) DEFAULT NULL,

`U\_address` varchar(100) DEFAULT NULL,

`U\_balance` decimal(10,2) DEFAULT NULL,

`U\_birthday` varchar(20) DEFAULT NULL,

`U\_headP` varchar(20) DEFAULT NULL,

PRIMARY KEY (`U\_id`,`U\_account`)

) ENGINE=InnoDB AUTO\_INCREMENT=20 DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of tb\_user

-- ----------------------------

INSERT INTO `tb\_user` VALUES ('1', '123567', 'mmd2', 'hhc', '男', '13655135769', '成都职业技术学院高新校区', '100.00', ' 1997.09.25', ' d1');

INSERT INTO `tb\_user` VALUES ('2', '585697', 'ccm', 'cao', '女', '13866912585', '成都市郫都区', '0.00', ' 1998.05.20', ' d2');

INSERT INTO `tb\_user` VALUES ('3', '612588', 'cck', 'wlk', '女', '13955368257', '成都市高新区天益二街', '50.00', '1899.09.23', 'd3');

INSERT INTO `tb\_user` VALUES ('4', '812567', 'k56', 'lig', '男', '13892856781', '高新区锦城大道', '500.00', ' 1900.06.15', ' d4');

INSERT INTO `tb\_user` VALUES ('5', '987568', 'll3', 'cch', '男', '13755612857', '高新区苏宁广场', '300.00', ' 1980.03.12', ' d5');

INSERT INTO `tb\_user` VALUES ('6', '555667', '555', 'xh', '女', '13688621357', '火车南站，凯德广场', '1000.00', ' 1986.10.17', ' d6');

INSERT INTO `tb\_user` VALUES ('7', '666338', '668', 'ful', '男', '13955321758', '成都南站', '2000.00', '1979.11.19', 'd7');

INSERT INTO `tb\_user` VALUES ('8', '925681', '777', 'yuw', '女', '13566289213', '华西医院，省体育馆', '3000.00', '1993.06.05', 'd8');

INSERT INTO `tb\_user` VALUES ('9', 'ffdshow', '123456', '无', '无', '无', '无', '0.00', '无', '无');

# 附录2 Android 端部分代码

1、网络请求部分代码

**import** android.util.Log;  
  
**import** com.example.findfood.app.LoginActivity;  
  
**import** java.io.ByteArrayOutputStream;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.io.OutputStream;  
**import** java.net.MalformedURLException;  
**import** java.net.URL;  
**import** java.net.URLEncoder;  
*/\*\*  
 \* Created by French on 2018/6/30.  
 \*   
 \* 网络请求代码  
 \*/***public class** HttpURLConnection {  
 **private static** String *LOGIN\_URL* = **"http://172.16.204.128:8080/XW\_servlet/"**;  
 **public static** String LoginByPost(String name, String data*/\*,String name/\*,String sex,String phone,String address,float balance\*/*) {  
 Log.*d*( LoginActivity.***TAG***, **"启动登录线程"**);  
 String msg = **""**;  
 **try** {  
 *//初始化URL* URL url = **new** URL(*LOGIN\_URL*+name);  
 java.net.HttpURLConnection conn = (java.net.HttpURLConnection) url.openConnection();  
 Log.*d*( LoginActivity.***TAG***, **"11111"**);  
 *//设置请求方式* conn.setRequestMethod(**"POST"**);  
 *//设置超时信息* conn.setReadTimeout(5000);  
 conn.setConnectTimeout(5000);  
 *//设置允许输入* conn.setDoInput(**true**);  
 *//设置允许输出* conn.setDoOutput(**true**);  
 *//post方式不能设置缓存，需手动设置为false* conn.setUseCaches(**false**);  
 *//我们请求的数据 请求时 & 这个必须有！！！* String data2= **"data="** + URLEncoder.*encode*(data, **"UTF-8"**);  
 Log.*d*( LoginActivity.***TAG***, **" ffff "** + data);  
 *//获取输出流* OutputStream out = conn.getOutputStream();  
 out.write(data2.getBytes());  
 out.flush();  
 out.close();  
 conn.connect();  
 **if** (conn.getResponseCode() == 200) {  
 *// 获取响应的输入流对象* InputStream is = conn.getInputStream();  
  
 Log.*d*( LoginActivity.***TAG***, **" ffff "** + is);  
 *// 创建字节输出流对象* ByteArrayOutputStream message = **new** ByteArrayOutputStream();  
 *// 定义读取的长度* **int** len = 0;  
 *// 定义缓冲区* **byte** buffer[] = **new byte**[1024];  
 *// 按照缓冲区的大小，循环读取* **while** ((len = is.read(buffer)) != -1) {  
 *// 根据读取的长度写入到os对象中* message.write(buffer, 0, len);  
 }  
 *// 释放资源* is.close();  
 message.close();  
 *// 返回字符串* msg = **new** String(message.toByteArray());  
 }  
  
 } **catch** (MalformedURLException e) {  
 e.printStackTrace();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 e.printStackTrace();  
 }  
 Log.*d*( LoginActivity.***TAG***, **"exit"**);  
  
 **return** msg;  
 }  
}

2、引导页部分代码展示

**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.v4.view.PagerAdapter;  
**import** android.support.v4.view.ViewPager;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.view.WindowManager;  
**import** android.widget.ImageView;  
**import** android.widget.LinearLayout;  
**import** android.widget.RelativeLayout;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
**import** com.example.findfood.R;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
*/\*\*  
 \* Created by French on 2018/6/28.  
 \* 启动页  
 \*/***public class** GuideActivity **extends** AppCompatActivity **implements** ViewPager.OnPageChangeListener, View.OnClickListener {  
 *// ViewPager的数据* **private** List<ImageView> **imageViewList**;  
 *// 点的组* **private** LinearLayout **llPointGroup**;  
 *// 选中的点view对象* **private** View **mSelectPointView**;  
 *// 开始体验按钮* **private** TextView **btnStartExperience**;  
 *// ViewPager* ViewPager **mViewPager**;  
 *// 点之间的宽度* **private int pWidth**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate( savedInstanceState );  
*// //去掉Activity上面的状态栏  
// getWindow().setFlags( WindowManager.LayoutParams. FLAG\_FULLSCREEN , WindowManager.LayoutParams. FLAG\_FULLSCREEN);  
  
// 状态栏透明* getWindow().setFlags( WindowManager.LayoutParams.***FLAG\_LAYOUT\_NO\_LIMITS***, WindowManager.LayoutParams.***FLAG\_LAYOUT\_NO\_LIMITS***);  
 setContentView( R.layout.***activity\_guide*** );  
 intiview();  
 }  
 **private void** intiview(){  
 **mViewPager** = (ViewPager) findViewById(R.id.***vp\_guide***);  
 **btnStartExperience** = (TextView) findViewById(R.id.***btn\_guide\_start\_experience***);  
 **llPointGroup** = (LinearLayout) findViewById(R.id.***ll\_guide\_point\_group***);  
 **mSelectPointView** = findViewById(R.id.***select\_point***);  
 initData();*// 初始化ViewPager数据* GuideAdapter adapter = **new** GuideAdapter();  
 **mViewPager**.setAdapter(adapter);  
 **mViewPager**.setOnPageChangeListener(**this**);*// 设置监听器* **btnStartExperience**.setOnClickListener(**this**);*// 按钮添加监听* }  
 *//* **private void** initData(){  
  
 **int**[] imageResIDs = {R.mipmap.***guid1***, R.mipmap.***guid2***,  
 R.mipmap.***guid3***};  
 **imageViewList** = **new** ArrayList<>();  
 ImageView iv;*// 图片* View view;*// 点* LinearLayout.LayoutParams params; *// 参数类* **for** (**int** i = 0; i < imageResIDs.**length**; i++) {  
 iv = **new** ImageView(**this**);  
 iv.setBackgroundResource(imageResIDs[i]);  
 **imageViewList**.add(iv);  
 *// 根据图片的个数, 每循环一次向LinearLayout中添加一个点* view = **new** View(**this**);  
 view.setBackgroundResource(R.drawable.***point\_normal***);  
 *// 设置参数* params = **new** LinearLayout.LayoutParams(30, 30);  
 **if** (i != 0) {  
 params.**leftMargin** = 10;  
 }  
 view.setLayoutParams(params);*// 添加参数* **llPointGroup**.addView(view);  
 }  
 }  
 *// 点击开始体验按钮之后* @Override  
 **public void** onClick(View view) {  
 Toast.*makeText*(getApplicationContext(), **"跳转"**, Toast.***LENGTH\_SHORT***).show();  
 System.***out***.println(**"页面跳转"**);  
 Intent intent=**new** Intent(GuideActivity.**this**,MainActivity.**class**);  
 startActivity(intent);  
 }  
 *//当页面正在滚动时 position 当前选中的是哪个页面 positionOffset 比例 positionOffsetPixels 偏移像素* @Override  
 **public void** onPageScrolled(**int** position, **float** positionOffset, **int** positionOffsetPixels) {  
  
 *//获取两个点间的距离,获取一次即可* **if**(**pWidth**==0) {  
 **pWidth** = **llPointGroup**.getChildAt(1).getLeft()  
 - **llPointGroup**.getChildAt(0).getLeft();  
 }  
  
 *// 获取点要移动的距离* **int** leftMargin = (**int**) (**pWidth** \* (position + positionOffset));  
 *// 给红点设置参数* RelativeLayout.LayoutParams params = (android.widget.RelativeLayout.LayoutParams) **mSelectPointView** .getLayoutParams();  
 params.**leftMargin** = leftMargin;  
 **mSelectPointView**.setLayoutParams(params);  
  
 }  
 *// 当页面被选中* @Override  
 **public void** onPageSelected(**int** position) {  
 *// 显示体验按钮* **if** (position == **imageViewList**.size() - 1) {  
 **btnStartExperience**.setVisibility(View.***VISIBLE***);*// 显示* } **else** {  
 **btnStartExperience**.setVisibility(View.***GONE***);*// 隐藏* }  
  
 }  
 *// 当页面滚动状态改变* @Override  
 **public void** onPageScrollStateChanged(**int** state) {  
  
 }  
 **class** GuideAdapter **extends** PagerAdapter {  
  
 @Override  
 **public int** getCount() {  
 **return imageViewList**.size();  
 }  
  
 @Override  
 **public boolean** isViewFromObject(View view, Object object) {  
 **return** view == object;  
 }  
 */\*  
 \* 删除元素  
 \*/* @Override  
 **public void** destroyItem(ViewGroup container, **int** position, Object object) {  
 container.removeView((View) object);  
 }  
  
 @Override  
 **public** Object instantiateItem(ViewGroup container, **int** position) {  
 ImageView iv = **imageViewList**.get(position);  
 container.addView(iv);*// 1. 向ViewPager中添加一个view对象* **return** iv; *// 2. 返回当前添加的view对象* }  
 }  
  
}