说明书

基于 Android 的菜谱查找分类 APP

目　录

1　需求分析

1．1　项目概述

基于 Android 的菜谱查找分类 APP 系统能够将传统的纸质菜谱存储在云端，实现对菜谱的分类查询，用户还能对菜品进行直接的查询。

本系统分为 Android 前端与服务器后端。服务器后端采用 Tomcat 搭建本地服务器，使用 Hibernate 框架，使后端程序更加友好。采用 servlet 实现后端服务程序。使用 json 进行数据交互，采用 HttpClient 工具包实现 Http 功能。

1．2　开发环境

服 务 器：Tomcat7

开发语言：Java、XML、SQL

开发工具：Android Studio

MyEclipse

MySQL

Navicat

1．3　项目需求

用户通过账号密码进行登录

系统发送验证码功能

用户通过账号及手机验证码进行注册

用户通过账号及手机验证码进行登录

用户通过账号及手机验证码进行重置密码操作

服务器对用户推送每日推荐功能

用户通过菜名直接查询菜谱及步骤

用户通过菜谱分类查看推荐菜系

用户通过推荐菜系查看菜品详细菜谱及步骤

用户通过自定义时间实现后台倒计时功能

2　概要设计

2．1　整体设计

2.1.1　服务器端设计

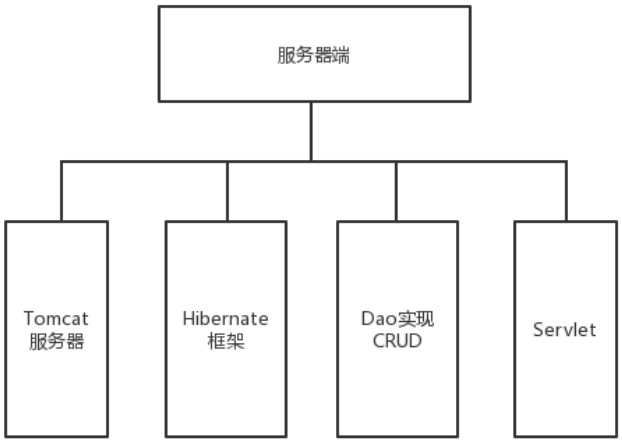

图 2.1 服务器端设计框架

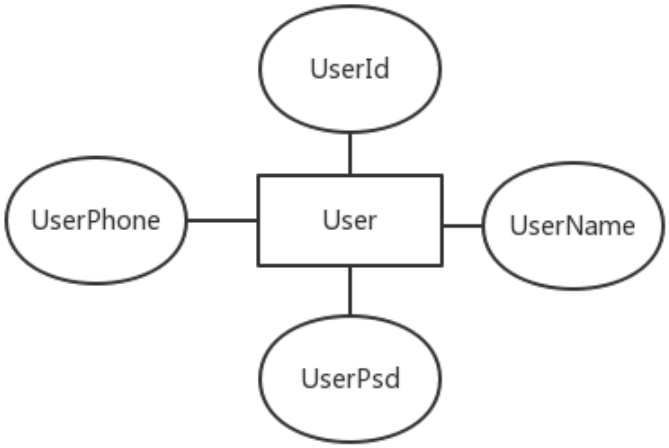

图 2.2 user 表设计

2.1.2 Android 前端设计

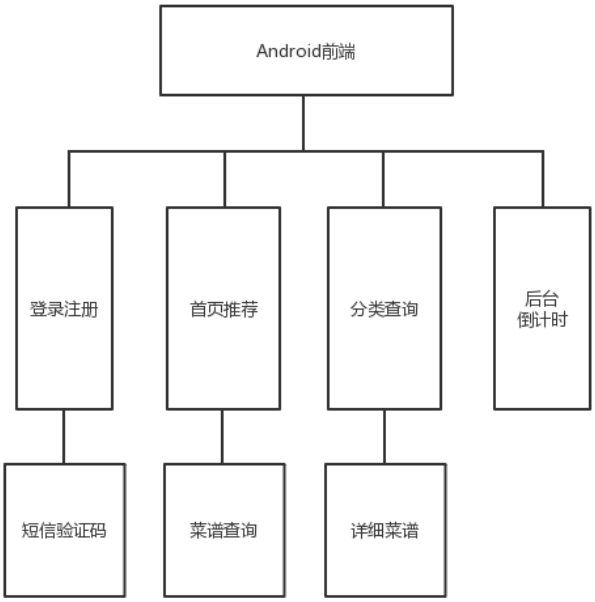

图 2.3 Android 前端设计框架

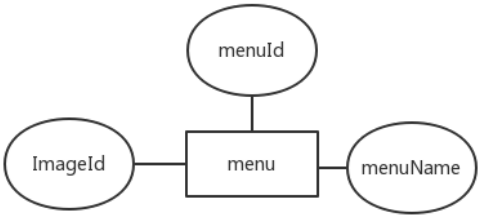

图 2.4 menu 类

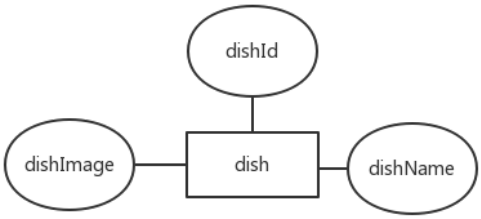

图 2.5 dish 类

图 2.6 steps 类

2．2　登录注册模块

2.2.1　功能设计

登录功能：前端将得到的数据通过网络请求发回到服务器端；服务器端对前端传回来的数据从数据库中进行查询，将查询结果通过 json 数据流返回前端。

注册功能：前端将得到的数据通过网络请求发回到服务器端；服务器端对前端传回来的数据从数据库中查询，若没有用户，进行插入数据操作，若用户存在，通过 json 数据流返回错误信息。

2.2.2　界面设计

图 2.7 登录端设计

图 2.8 注册端设计

2．3　短信验证码模块

2.3.1　功能设计

采用外部接口，Android 端生成六位随机数，生成规定的字符串发送网络请求，同时将六位随机数返回。

此时用户接收到短信验证码，输入时与这六位随机数比较，成功则进行短信验证码登录、注册、修改密码等操作，若不成功则六位数作废，用户需重新申请短信验证码。

2.3.2　界面设计

图 2.3.1 为修改密码原型设计页面，图 2.3.2 为短信验证码登录原型设计页面。

图 2.9 登录端设计

图 2.10 注册端设计

2．4　每日推荐及搜索框模块

2.4.1　功能设计

每日推荐页面获取本地服务器内的推送内容，用户点击会实现查看该菜谱的内容。搜索框输入菜名，会出现完整菜谱。

2.4.2　界面设计

图 2.11 每日推荐页面设计

2．5　分类查询模块

2.5.1　功能设计

分类页面实现菜品的分类，点击单个分类出现多个该分类里的菜品，再对单个菜品进行点击，能出现菜谱的详细信息及步骤。

2.5.2　界面设计

图 2.12 菜品分类页面设计

2．6　倒计时设计模块

2.6.1　功能设计

实现后台倒计时功能，进入其他页面时，也在后台运行，在实际的生活场景中能进行良好的应用。

2.6.2　界面设计

图 2.13　倒计时界面设计

3　详细设计

3．1　系统功能描述

3.1.1　服务器端功能描述

第一步 通过 Hibernate 连接数据库

第二步 写 DAO 包，成功实现 CRUD 操作

第三步 通过 Servlet，启动服务器后成功实现 CRUD 操作

第四步 生成 json，访问 Servlet 成功显示

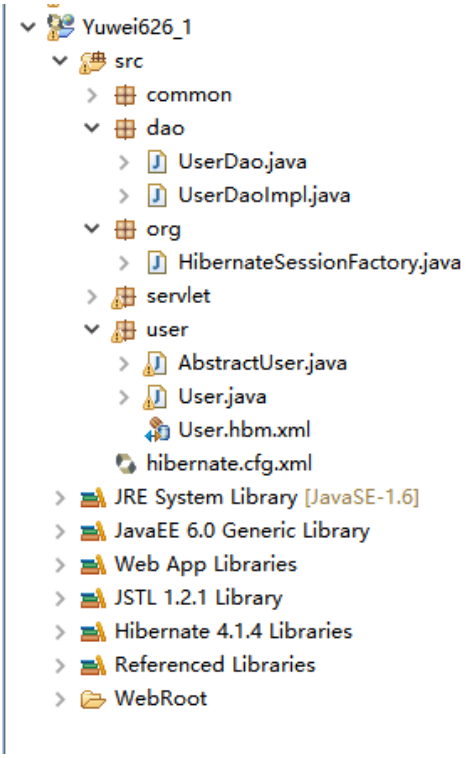

图 3.1 MyEclipse 框架

3.1.2　Android 前端功能描述

通过 HttpClient 接收服务器端传回的 json，进行对数组等的解析，成功在客户端实现功能。通过访问本机 IP 地址，访问服务器里的文件并读取后进行显示。

采用 CallBack 函数对子线程里的数据进行回调。采用 Handler 方法成功在子线程执行完后改变 UI 中的内容。

3．2　登录注册模块

3.2.1　功能描述

Android 前端：通过用户输入的信息通过本地服务器地址生成 URL，发网络请求去服务器端查询并返回 json，对 json 进行解析可得到是否符合，再进行登录入主界面或者重新输入操作。

服务器后端：得到客户端发来的数据，进入数据库进行操作，将返回结果的 json 字符串并发送给前端进行逻辑判断。

3.2.2　界面展示

图 3.2 登录界面

图 3.3 注册界面

3．3　短信验证码模块

3.3.1　功能描述

通过(int)((Math.random()*9+1)*100000)生成六位随机数进行发送。
通过以下函数进行 MD5 加密。

```
md5 = MessageDigest.getInstance("MD5");
byte[] bytes = md5.digest(s.getBytes());
String result = "";
for (byte b : bytes) {
    String temp = Integer.toHexString(b & 0xff);
    if (temp.length() == 1) {
        temp = "0" + temp;
    }
    result += temp;
}
```

3.3.1　界面展示

图 3.4 登录界面

图 3.5 注册界面

3．4　每日推荐及搜索框模块

3.4.1　欢迎页功能描述

访问本地服务器里的内容进行显示的代码如下：

```
String ADDRESS = "http://10.0.161.102:8080/img/";
String url1 = ADDRESS +"11091_451805.jpg";
```

3.4.2　欢迎页界面展示

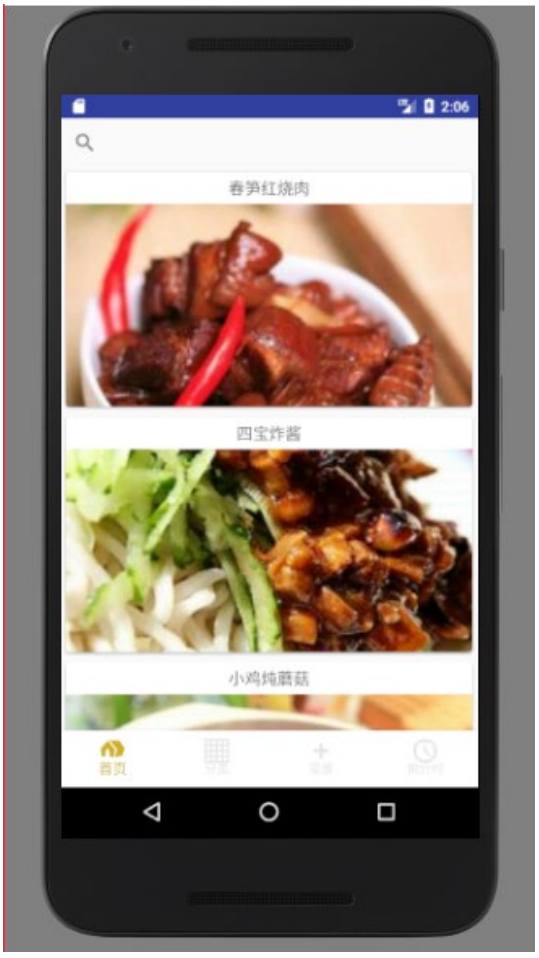

图 3.6 推荐主界面

图 3.7 菜谱界面

3.4.3　搜索框功能描述

搜索框的主要代码如下：

```
searchView.setIconified(true);
Log.e(TAG, "onQueryTextSubmit: "+ query );
Intent intent = new Intent(getContext(),OneDish.class);
intent.putExtra("DISH_NAME",query);
startActivity(intent);
return true;
```

3.4.4　搜索框界面展示

图 3.8 搜索界面

图 3.9 搜索成功菜谱界面

3．5　分类查询模块

3.5.1　功能描述

采用 Recyclerview 进行界面的设置，采用 Glide 从指定地址获取图片

```
Glide.with(context).load(menu.getImageId()).into(holder.menu_image);
holder.menu_name.setText(menu.getMenu_name());

int position = viewHolder.getAdapterPosition();
MainFragment_2.Menu menu = menuList.get(position);
Toast.makeText(v.getContext(),"you cliked view" +
menu.getMenu_name(),Toast.LENGTH_SHORT).show();
Intent intent = new Intent(v.getContext(),MainFragment_2_ListMenu.class);
intent.putExtra("id",menu.getMenu_id());
startActivity(v.getContext(),intent,new Bundle());
```

3.5.2 界面展示

图 3.10 菜谱分类界面

3.5.3 分类菜单点击查询

将查询到的菜加入 List 中，在 Handler 中执行完成后进行页面设置

```
JSONObject object1 = new JSONObject(result);
String data = object1.getString("data");
JSONArray jsonArray = new JSONArray(data);
JSONObject object3 = jsonArray.getJSONObject(0);
final String id = object3.getString("id");
final String title = object3.getString("title");
String albums = object3.getString("albums");
JSONArray jsonArray1 = new JSONArray(albums);
final String albums1 = jsonArray1.getString(0);


Dish dish = new Dish(id, title, albums1);
dishList.add(dish);
```

```
adapter = new
MainFragment_2_ListMenuAdapter(MainFragment_2_ListMenu.this,dishList);
recyclerView.setAdapter(adapter);
adapter.notifyDataSetChanged();
```

3.5.4 点击界面展示

图 3.11 菜谱分类界面

图 3.12 菜谱分类界面

3．6 倒计时设计模块

3.6.1 功能描述

```
min = minute.getText().toString();
s = second.getText().toString();
int count_min =Integer.parseInt(min);
int count_s =Integer.parseInt(s);
int AllTime = (count_min*60+count_s)*1000;
mc = new Mycount(AllTime,1000);
mc.start();
```

3.6.2 界面展示

图 3.13 后台计时界面

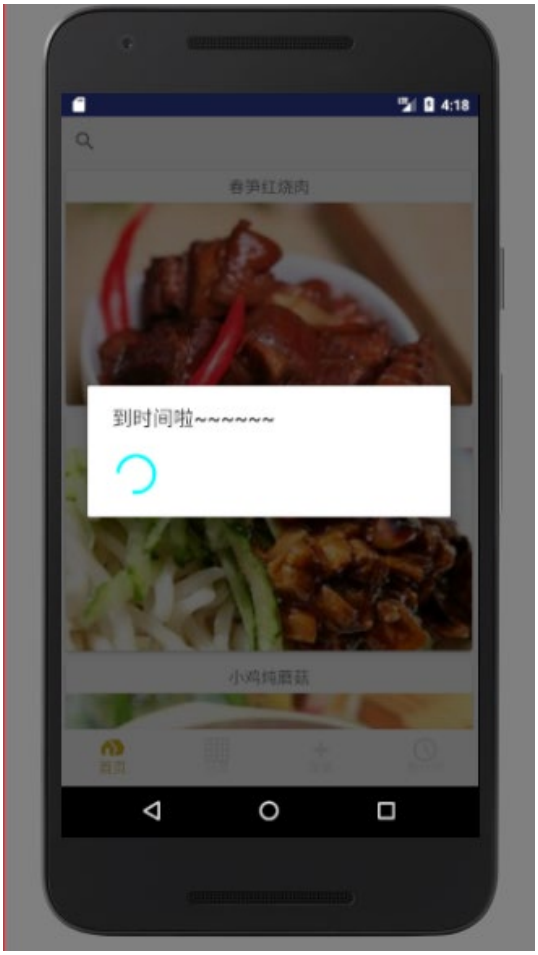

图 3.14 后台计时成功界面

4 软件测试

软件测试（英语：Software Testing），描述一种用来促进鉴定软件的正确性、完整性、安全性和质量的过程。

白盒测试又称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。白盒测试是一种测试用例设计方法，盒子指的是被测试的软件，白盒指的是盒子是可视的，你清楚盒子内部的东西以及里面是如何运作的。"白盒"法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。"白盒"法是穷举路径测试。在使用这一方案时，测试者必须检查程序的内部结构，从检查程序的逻辑着手，得出测试数据。贯穿程序的独立路径数是天文数字。

黑盒测试也称功能测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。

灰盒测试，是介于白盒测试与黑盒测试之间的一种测试，灰盒测试多用于集成测试阶段，不仅关注输出、输入的正确性，同时也关注程序内部的情况。灰盒测试不像白盒那样详细、完整，但又比黑盒测试更关注程序的内部逻辑，常常是通过一些表征性的现象、事件、标志来判断内部的运行状态。

此次我采用白盒测试法对前端、后端程序进行测试，看软件测试结果是否与预期效果一致。

4．1　服务器端测试结果

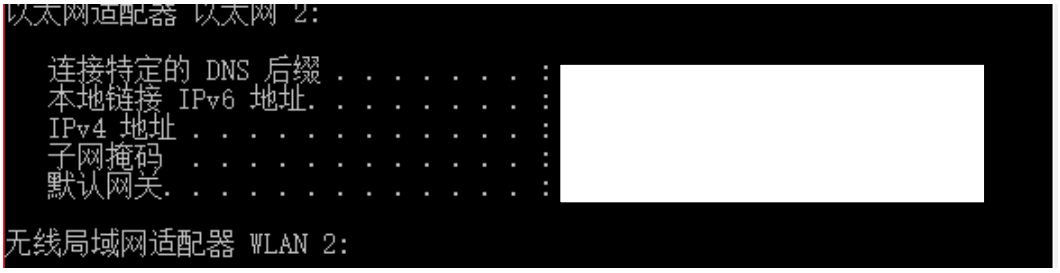

图 4.1 本地 IP 地址

图 4.2 注册成功页面

图 4.3 注册时账号已存在页面

图 4.4 登陆成功页面

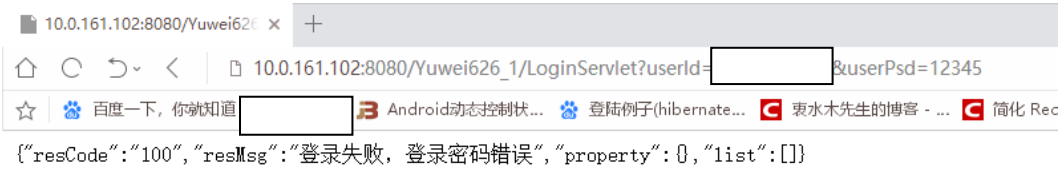

图 4.5 登陆失败页面

4．2　Android 前端测试结果

4.2.1　登录注册测试结果

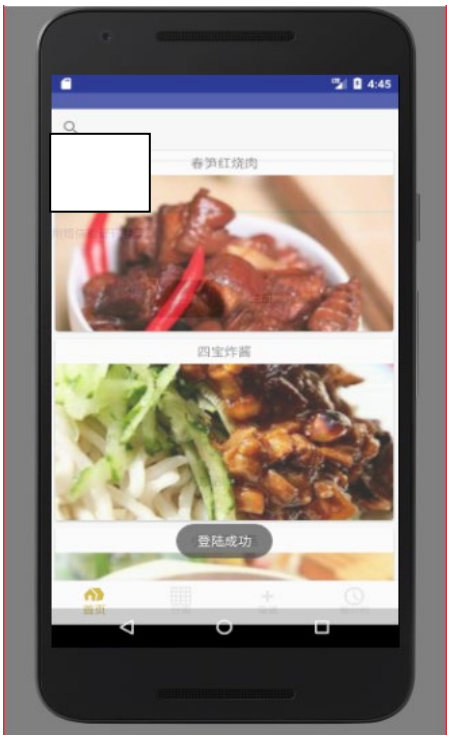

图 4.6 登陆成功页面

图 4.7 登陆失败页面

图 4.8 注册失败页面

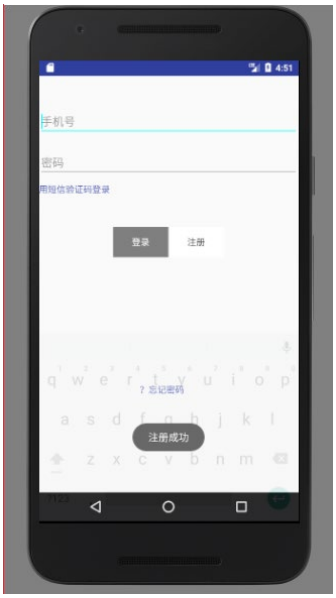

图 4.9 注册成功页面

图 4.10 账号存在页面

4.2.2 验证码测试结果

图 4.11 验证码登录页面

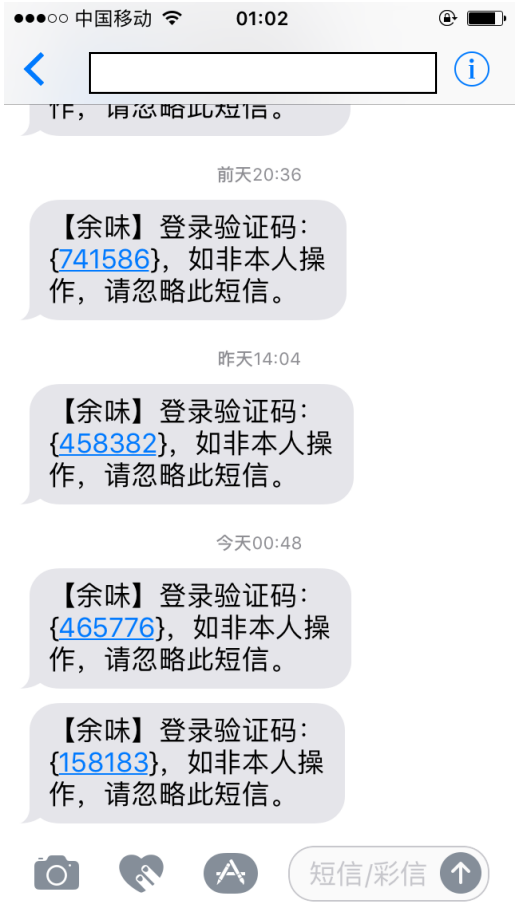

图 4.12 手机接收验证码页面

4.2.3 每日推荐及搜索框测试结果

图 4.13 欢迎页面点击菜谱

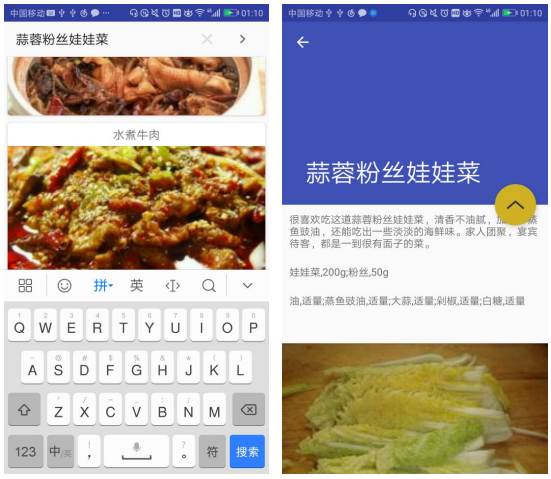

图 4.14 搜索框成功页面

4.2.4　分类查询测试结果

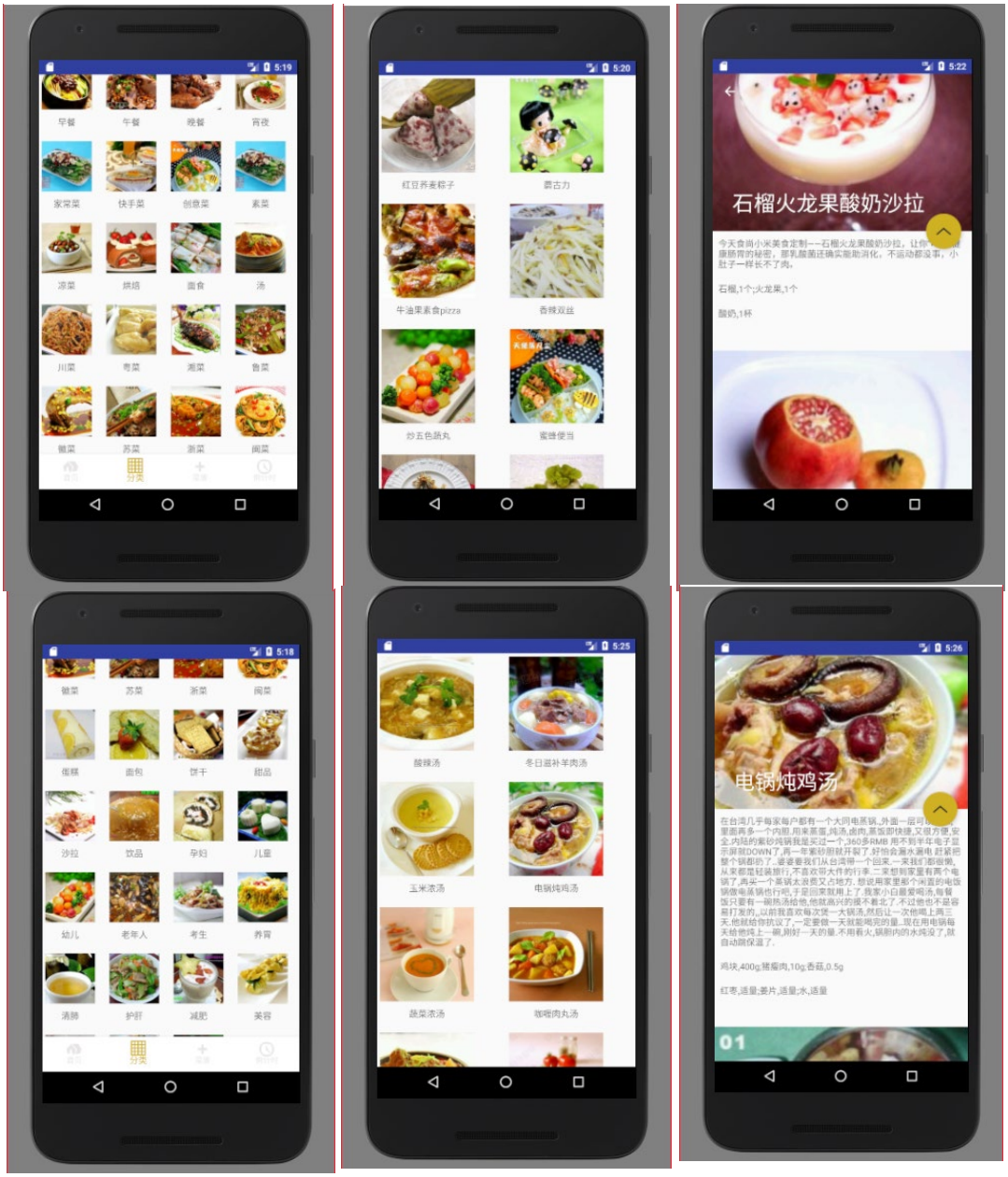

图 4.15 分类检索页面

4.2.5　倒计时功能测试结果

图 4.16 后台定时器

图 4.17 后台定时器成功

图 4.18 后台定时器暂停

5 结论

通过本次的课设，自己搭建了服务器之后，发现学长学姐口中的自己搭建服务器也只是纸老虎。了解到了现在常用的轻量级数据传输格式 json，在调用短信验证码的 API 接口时，了解到了 MD5 加密还有随机数算法。子线程中进行网络请求时，学会使用了 Handler 以及 CallBack 函数。

参 考 文 献

[1] 张海藩.软件工程导论（第六版）.北京：清华大学出版社.

[2] 郭霖.第一行代码（第二版）.北京：人民邮电出版社.

[3] 郑阿奇.JavaEE 实用教程（第二版）.北京：电子工业出版社.

[4] 丁毓峰 毛雪涛.JavaWeb 开发教程（第六版）.北京：人民邮电出版社.

致　谢