

**课程设计报告**

**课程名称:**  移动计算技术与应用

**设计题目：** 掌上菜谱

**院 系：** 计算机科学与工程学院

**专 业：** 　 软件工程

**班　　级：** 单招141

**学 号：** 092214109

**姓 名：** 张立飞

**指导教师：** 孙 霞

**设计地点：** N6-113

**开课时间：** 2016 至 2017 学年第 2 学期

**目录**

目录

[1. 引言 4](#_Toc484440392)

[1.1题目描述 4](#_Toc484440393)

[1.2设计意义 4](#_Toc484440394)

[2. 开发环境 4](#_Toc484440395)

[2.1 硬件环境 4](#_Toc484440396)

[2.2 软件环境 4](#_Toc484440397)

[3. 相关技术及知识点 4](#_Toc484440398)

[3.1 滑动控件的使用 4](#_Toc484440399)

[3.1.1. 新建界面资源文件，添加滑动控件资源 4](#_Toc484440400)

[3.1.2. 在Activity中为控件添加图片 5](#_Toc484440401)

[3.1.3. 切换导航点 5](#_Toc484440402)

[3.2 滑动控件的定时滑动 5](#_Toc484440403)

[3.3 适配器的使用 6](#_Toc484440404)

[3.3.1 定义一个适配器 6](#_Toc484440405)

[3.3.2 使用一个适配器 6](#_Toc484440406)

[3.4 页面的跳转与传值 6](#_Toc484440407)

[3.4.1 页面的跳转和传值 6](#_Toc484440408)

[3.4.2 数值的获取 7](#_Toc484440409)

[3.5 评分控件的使用 7](#_Toc484440410)

[3.5.1 添加一个评分控件 7](#_Toc484440411)

[3.5.2 为评分控件设置参数 7](#_Toc484440412)

[3.6 全局对象的使用 7](#_Toc484440413)

[3.6.1 定义全局类 7](#_Toc484440414)

[3.6.2 设置全局参数 7](#_Toc484440415)

[3.7 监听器的使用 8](#_Toc484440416)

[3.8 Scroll View控件的一些处理 8](#_Toc484440417)

[4. 需求分析 8](#_Toc484440418)

[4.1 系统界面需求分析 8](#_Toc484440419)

[4.2 系统性能需求分析 9](#_Toc484440420)

[4.3系统功能需求分析 9](#_Toc484440421)

[4.4 系统流程图 9](#_Toc484440422)

[5. 设计与实现 10](#_Toc484440423)

[5.1 界面设计与实现 10](#_Toc484440424)

[5.1.1 首页 10](#_Toc484440425)

[5.1.2 搜索界面 12](#_Toc484440426)

[5.1.3 信息界面 13](#_Toc484440427)

[5.1.4 食谱界面 15](#_Toc484440428)

[5.2 全局对象的设计 17](#_Toc484440429)

[5.2.1 全局对象的设计 17](#_Toc484440430)

[5.2.2 全局对象的使用 17](#_Toc484440431)

1. 引言

1.1题目描述

开发一个掌上菜谱应用，收录名菜和美食做法，提供菜谱分类、搜索等功能，方便用户找到想要的各式菜品，并能提供健康饮食指导建议。且必须完成完成主要功能，注意功能完备、界面美观。在右上角的菜单出有一个“关于”菜单功能，显示开发者的学号，姓名，程序功能介绍，程序版本。具备两个以上的activity切换。进行某些操作时给出提示信息和背景音乐。能够i存储轻量级数据。若使用SQLite，则要求能够进行记录的添加、修改、删除、读取等操作。

1.2设计意义

通过学习，掌握Android移动应用程序开发的基本技术，能开发具有系统标准的实用程序及简单的网络程序。为以后从事该方向深入学习研究做好准备。

2. 开发环境

2.1 硬件环境

Cpu : Intel Core i7 4700HQ @2.40GHZ

Mainboard : ASUSTek COMPUTER INC X450JN

Memory : DDR3 8GBytes

Graphics : Intel HD Graphics 4600

Graphics : NVIDIA GeForce 840M

2.2 软件环境

Coding IDE : IntelliJ IDEA 2017.1

Java SDK : 8u111

Android SDK : 25.06

Design : starUml v2.80

3. 相关技术及知识点

3.1 滑动控件的使用

3.1.1. 新建界面资源文件，添加滑动控件资源

代码段 1-滑动控件资源

<android.support.v4.view.ViewPager

*android:layout\_width*="match\_parent"

*android:layout\_height*="wrap\_content" />

<LinearLayout *android:layout\_width*="match\_parent"

*android:layout\_height*="20dp"

*android:orientation*="horizontal">

</LinearLayout>

3.1.2. 在Activity中为控件添加图片

代码段 2-添加ViewPager的图片资源

for(int i = 0; i < imgs.length; i++){

ImageView iv = new ImageView(this);

LayoutParams params = new LayoutParams(LinearLayout.LayoutParams.MATCH\_PARENT, LinearLayout.LayoutParams.MATCH\_PARENT);

iv.setLayoutParams(params);

iv.setScaleType(ScaleType.FIT\_CENTER);

iv.setImageResource(imgs[i]);

list.add(iv);

}

3.1.3. 切换导航点

代码段 3-设置监听器切换导航点

viewpager.setOnPageChangeListener(new OnPageChangeListener() {

@Override

public void **onPageSelected**(int arg0) {

for(int i = 0; i < list.size(); i++) {

ImageView iv = (ImageView) layout.getChildAt(i);

if(i == arg0) {

iv.setImageResource(R.drawable.page\_on);

}

else {

iv.setImageResource(R.drawable.page\_off);

}

}

if(arg0 == list.size() - 1) {

setTimer();

}

}

});

3.2 滑动控件的定时滑动

代码段 4-定时滑动

Timer timer = new Timer();

TimerTask timerTask = new TimerTask() {

public void **run**() {

handler.sendMessage(handler.obtainMessage(1));

}};

timer.schedule(timerTask, 0, 1000 \* 2);

3.3 适配器的使用

3.3.1 定义一个适配器

代码段 5-定义一个适配器

public class **FoodAdapter** extends **BaseAdapter** {

protected Context context;

protected LayoutInflater inflater;

protected int resource;

protected int[] list;

public **FoodAdapter**(Context context, int resource, int[] list){

inflater = LayoutInflater.from(context);

this.context = context;

this.resource = resource;

}

@Override

public int **getCount**() {

return 10;

}

@Override

public Object **getItem**(int position) {

return list[position];

}

@Override

public long **getItemId**(int position) {

return position;

}

@Override

public View **getView**(int position, View convertView, ViewGroup parent) {

}

3.3.2 使用一个适配器

代码段 6-使用一个适配器

Type adapter = new Type(this, R.layout.layout, list);

this.listView.setAdapter(adapter);

3.4 页面的跳转与传值

3.4.1 页面的跳转和传值

代码段 7-Activity之间的传值

final Intent intent = new Intent(Activity.this, Activity.class);

intent.putExtra("keyWords", keyWords);

startActivity(intent);

3.4.2 数值的获取

代码段 8-获取Activity的传值

this.index = getIntent().getIntExtra("id", 0);

（传值类型与获取类型必须相同）

3.5 评分控件的使用

3.5.1 添加一个评分控件

代码段 9-添加评分控件资源

<RatingBar

*android:padding*="5dp"

*android:layout\_width*="match\_parent"

*android:layout\_height*="0dp"

*android:layout\_weight*="1"

*android:numStars*="3"

*android:stepSize*="0.5"

*android:layout\_gravity*="center"/>

3.5.2 为评分控件设置参数

代码段 10-设置评分控件参数

this.ratingBar.setMax(Max);

this.ratingBar.setRating(rating);

3.6 全局对象的使用

3.6.1 定义全局类

代码段 11-定义全局类

public class **Data** extends **Application** {

@Override

public void **onCreate**(){

super.onCreate();

}

3.6.2 设置全局参数

代码段 12-添加全局对象

<application

*android:name*=".ToolBean.Data">

</application>

3.7 监听器的使用

代码段 13-为EditText设置监听器

this.searchEdit.setOnKeyListener(new View.OnKeyListener() {

@Override

public boolean **onKey**(View v, int keyCode, KeyEvent event){

return false;

}

});

3.8 Scroll View控件的一些处理

如果直接来ListView控件上套用ScrollView的话，界面会出现只出现一个Item，后来查阅资料后发现，是因为ScrollView无法预判ListView控件中有多少个内容，导致无法计算要滑动区域的大小。所以需要对ScrollView区域的内容进行重新计算。

代码段 14-刷新的ScrollView高度

setListViewHeightBasedOnChildren(listView, adapter);

((ScrollView) findViewById(R.id.scrollView)).smoothScrollTo(0,20);

private void **setListViewHeightBasedOnChildren**(ListView listView, ListAdapter listAdapter) {

int totalHeight = 0;

for (int i = 0; i < listAdapter.getCount(); i++) {

View listItem = listAdapter.getView(i, null, listView);

listItem.measure(0, 0);

totalHeight += listItem.getMeasuredHeight();

}

ViewGroup.LayoutParams params = listView.getLayoutParams();

params.height = totalHeight + (listView.getDividerHeight() \* (listAdapter.getCount() - 1));

((ViewGroup.MarginLayoutParams)params).setMargins(15, 15, 15, 15);

listView.setLayoutParams(params);

}

4. 需求分析

4.1 系统界面需求分析

标明需求中具有收录名菜和美食做法，提供菜谱分类、搜索等功能，方便用户找到想要的各式菜品，并能提供健康饮食指导建议。

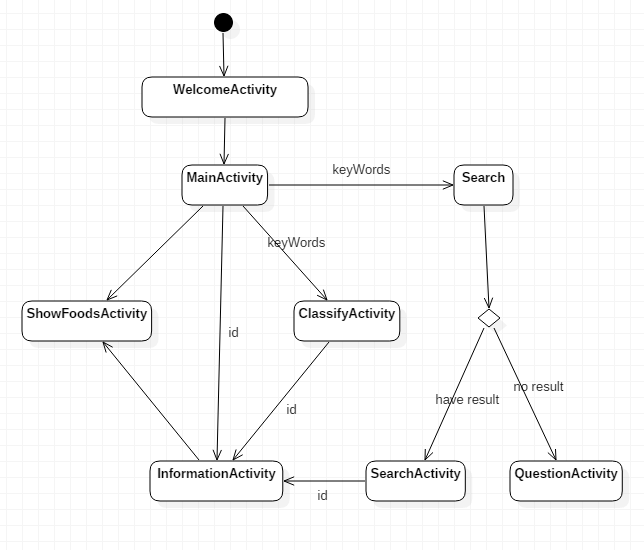
所以分析界面需要有首页，用于展示所有收录的名菜和美食做法。每个菜品为ListView中的一个item，点击后能够查看到此菜的做法。还需要有分类界面，将美食分为各个菜系，每个菜系分别显示各个菜系的美食；同样，每个菜品点进能够查看每个美食的做法。还需要有搜索界面或搜索框用于搜索菜品或菜系信息。还需要有一个提供健康饮食指导建议的界面。

4.2 系统性能需求分析

综合考虑各方面因素，在本应用中不使用SQLite，所有的数据以数组的方式封装在全局对象Application中，这当中包括，菜品图片、菜品做法信息文件列表、卡路里数值，健康度数值，饱腹度数值。当使用到这些数据时，先从系统中提取application对象，然后通过标识的方式获取数据信息。

4.3系统功能需求分析

* 显示所有的菜品
* 提供搜索功能
* 提供分类功能
* 提供添加菜品功能
* 提供菜品点赞功能
* 提供美食建议

4.4 系统流程图

图片 1-系统Activity跳转图

WelComeActivity to MainActivity ：从欢迎页到首页，当欢迎页的页面到最后一页的时候过三秒自动跳转到首页。

MainActivity to Search : 从首页到查找，在首页中输入关键字点击查询按钮，系统自动为其查找数据，如果查找了相关资料就跳转到SearchActivity，如果没有找到就跳转到QuestionActivity。

MainAcitivity to ClassifyActivity : 从首页到分类页面，当在首页中没有输入任何关键字就点击跳转会直接跳转到分类界面。

MainActivity to InformationActivity : 从首页到信息页面，当在首页中点击了某一个美食，就跳转到信息界面，显示美食信息。

ClassifyActivity to InformationActivity : 从分类页面到信息页面，当在分类页面点击了某个食物就跳转到信息页面。

InformationActivity to ShowFoodsActivity : 从信息界面可以点击菜单栏跳转到食谱页面。

5. 设计与实现

5.1 界面设计与实现

图片 2-系统首页设计图

5.1.1 首页

当系统跳出欢迎页就跳转到首页当中。首页从上至下，从左至右分别是：滑动控件，输入控件，按钮，列表控件。

* 滑动控件：滑动控件用于动态显示图片列表。详细信息见知识点。
* 输入控件：输入控件用于输入系统搜索关键词。当此页面初始化时，默认的文字为搜索字样，颜色为灰色。当点击触摸到控件时，控制文字为空，且颜色变黑，能自动跳出输入法。

代码段 15-设置输入控件的监听器

this.searchEdit.setOnTouchListener(new View.OnTouchListener() {

@Override

public boolean **onTouch**(View v, MotionEvent event) {

searchEdit.setText("");

searchEdit.setFocusable(true);

searchEdit.setFocusableInTouchMode(true);

searchEdit.requestFocus();

searchEdit.setTextColor(Color.rgb(92, 92, 92));

InputMethodManager inputMethodManager = (InputMethodManager)getSystemService(Context.INPUT\_METHOD\_SERVICE);

inputMethodManager.toggleSoftInput(0, InputMethodManager.HIDE\_NOT\_ALWAYS);

return true;

}

});

* 按钮：按钮用于查询信息操作，当用户点击此按钮时，系统调用方法，先获取输入控件中的数据，然后判断如果当前输入框中为空，则跳转到分类界面，如果不为空，则先保存关键词，然后跳转到查询页面，由后面页面自行处理。

代码段 16-查询预备操作

String keyWords = searchEdit.getText().toString();

keyWords.trim();

if(keyWords.equals("搜索")) {

keyWords = "";

}

if(keyWords.length() != 0) {

final Intent intent = new Intent(MainActivity.this, SearchActivity.class);

intent.putExtra("keyWords", keyWords);

startActivity(intent);

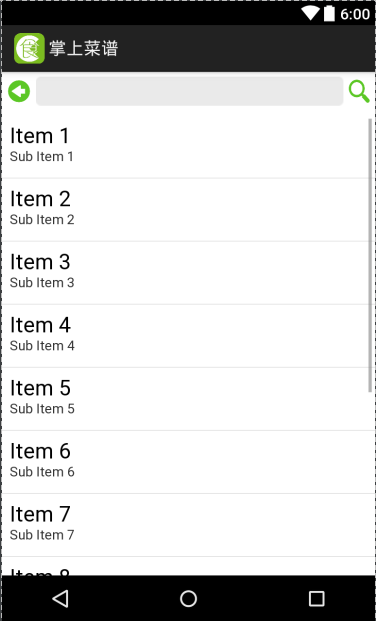
} else {

final Intent intent = new Intent(MainActivity.this, ClassifyActivity.class);

startActivity(intent);

}

* 列表控件：用于显示美食列表。详细信息见知识点。

5.1.2 搜索界面

图片 3-搜索界面

搜索界面控件由上至下，由左至右分别是返回按钮，搜索输入框，查询按钮，下拉列表框。

* 返回按钮：返回按钮用于从当前页面返回到首页。
* 搜索输入框：用于显示搜索内容。当用户在首页输入查询关键词，点击查询跳转到此页面后，先将关键词显示在输入框中，然后系统查询跟关键词有关的美食显示在下方列表中。为其添加了相关监听器，详细见代码段15
* 查询按钮：点击查询后，获取输入框中的内容，将其发送给本页面，然后跳转，交给本页面的下一个引用处理。

代码段 17-跳转查询

String keyWords = this.editText.getText().toString();

keyWords.trim();

if(keyWords.length() != 0) {

Intent intent = new Intent(SearchActivity.this, SearchActivity.class);

intent.putExtra("keyWords", keyWords);

startActivity(intent);

}

* 列表控件：用于显示美食列表。详细信息见知识点。

5.1.3 信息界面

图片 4-信息界面

本页面用于显示美食的信息，用户可以点击左边的赞按钮或点击右边的添加到菜谱，将这个美食添加到今日菜谱。控件由上至下分别是：图片控件，文字控件，文字控件，图片控件。

* 图片控件：此控件用于显示美食的图片。
* 文本控件：上面一个文本控件用于显示美食的名字，下面一个文本控件用于显示美食的做法。
* 点赞按钮：用于对美食点赞，点击后对旁边的文字按钮加一。并对其进行控制。如果今天已经点赞过则无法继续点赞给出提示。

代码段 18-点赞控制

if(!this.isgood) {

int number = Integer.parseInt(textView3.getText().toString());

textView3.setText(++number + "");

((ImageView) findViewById(R.id.imageView2)).setImageResource(R.drawable.good\_after);

this.isgood = true;

this.application.setIsGood(this.index);

} else {

Toast.makeText(InformationActivity.this, "你今天已经赞过了！", 2000).show();

}

* 添加按钮：用于将美食添加新今日菜谱，每修改一次就将其文本内容改变。

代码段 19-添加食谱控制

if(!this.isadd) {

Toast.makeText(InformationActivity.this, "已加入今日菜谱",2000).show();

this.isadd = true;

this.application.setIsAdd(this.index);

this.application.addFood(this.index);

((TextView) findViewById(R.id.textView6)).setText("移除菜谱");

((ImageView) findViewById(R.id.imageView4)).setImageResource(R.drawable.delete);

} else {

Toast.makeText(InformationActivity.this, "已移除今日菜单", 2000).show();

this.isadd = false;

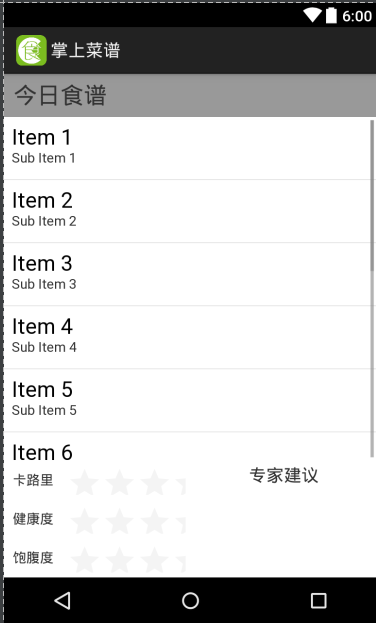
this.application.removeIsAdd(this.index);

this.application.removeFood(this.index);

((TextView) findViewById(R.id.textView6)).setText("添加到菜谱");

((ImageView) findViewById(R.id.imageView4)).setImageResource(R.drawable.add);

}

5.1.4 食谱界面

图片 5-食谱界面

本页面用于显示今日添加进的食谱，可以看到系统对用户的评分并给出专家的建议。由上至下的控件为文本显示控件、列表控件，评分控件，文本控件。

* 列表控件：用于显示今日点的菜谱。详细见知识点。
* 评分控件：用于显示系统对今日食谱的评分。先从系统中获取分值列表，然后循环计算出每个食物的质量和，然后求取平均值，因为标准的质量的区间为[0,10]，所以评分控件的Max为10，然后计算分值并显示。

double sum\_karuly = 0;

double sum\_healty = 0;

double sum\_satety = 0;

for(int i = 0; i < this.application.getIsExist(); ++i) {

sum\_karuly += this.karulyList[this.application.getChoice()[i]];

sum\_healty += this.healtyList[this.application.getChoice()[i]];

sum\_satety += this.satietyList[this.application.getChoice()[i]];

}

this.ratingBar1.setMax(10);

this.ratingBar2.setMax(10);

this.ratingBar3.setMax(10);

sum\_karuly = sum\_karuly \* 1.0 / this.application.getIsExist();

sum\_healty = sum\_healty \* 1.0 / this.application.getIsExist();

sum\_satety = sum\_satety \* 1.0 / this.application.getIsExist();

this.ratingBar1.setRating((float) sum\_karuly \* 3 / 8);

this.ratingBar2.setRating((float) sum\_healty \* 3 / 8);

this.ratingBar3.setRating((float) sum\_satety \* 3 / 8);

代码段 20-评分操作

* 文本控件：用于显示用户的文本提示信息。在此使用文本拼接的方法，将每个选集的评价链接起来。

代码段 21-连接评价信息

String information = new String();

information += this.getKarulySuggest(sum\_karuly);

information += "\n";

information += this.getHealtySuggest(sum\_healty);

information += "\n";

information += this.getSatetySuggest(sum\_satety);

代码段 22-文本评价示例

private String **getKarulySuggest**(double number) {

switch ((int)number)

{

case 1 : case 2 : case 3 : return "这个餐谱的卡路里还不能满足您的消耗呢！";

case 4 : case 5 : case 6 : return "您可以再添加一些食物哦！";

case 7 : case 8 : case 9 : return "这个餐谱卡路里太高了，你要攒肉肉吗？";

case 10 :case 0: return "？";

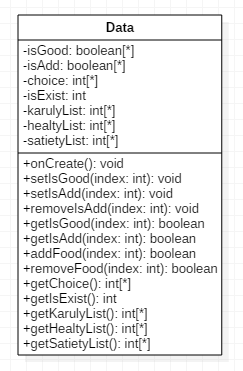
default: return "";

}

}

5.2 全局对象的设计

由于本项目没有使用SQLite，也没有网络接口资源。所有的资源通过我从网上获取，并保存在本地。在项目使用的过程中，需要容器来保存系统中的整个数据来贯穿使用。所以通过相关的查询，设计了一个全局的对象。

5.2.1 全局对象的设计

图片 6-全局类设计

属性private boolean[] isGood; 用于保存某个美食今日是否已被点赞

属性private boolean[] isAdd; 用于判断某个食物今日是否已经添加到菜单

属性private int[] choice; 用于保存今日所有菜单的选项

属性private int isExist; 用于保存今日菜单内的数据的个数

属性private int[] karulyList; 用于保存卡路里值

属性private int[] healtyList; 用于保存健康值

属性private int[] satietyList; 用于保存饱腹值

方法public void onCreate() 系统构造的自动重写的方法

方法public void setIsGood(int index) 设置某个美食被点赞

方法public void setIsAdd(int index)设置某个美食被添加了

方法public void removeIsAdd(int index)去除某个被添加的美食

方法public boolean getIsGood(int index)获取某个美食是否被点赞

方法public boolean getIsAdd(int index)获取某个美食是否被添加

方法public boolean addFood(int index)添加某个美食

方法public boolean removeFood(int index)去除某个被添加的美食

方法public int[] getChoice()获取选择列表

方法public int getIsExist()获取已经添加的书目

方法public int[] getKarulyList()获取卡路里列表

方法public int[] getHealtyList()获取健康值列表

方法public int[] getSatietyList()获取饱腹值列表

5.2.2 全局对象的使用

配置全局变量

代码段 23-配置全局对象

<application

android:allowBackup="true"

android:icon="@mipmap/icon"

android:label="@string/app\_name"

android:supportsRtl="true"

android:theme="@style/AppTheme"

android:name=".ToolBean.Data">

</application>

获取全局对象

代码段 24-获取全局对象

this.application = (Data) getApplication();

**参考资料**

[1] 李刚. 疯狂 Android 讲义[M]. 北京：电子工业出版社，2013:25-42.

[2] 杨丰盛. Android 技术内幕[M]. 北京： 机械工业出版社，2011:77-89.

[3] 杨云君. Android 的设计与实现[M]. 北京：机械工业出版社，2013:45-49.

[4] 柯元旦. Android 内核剖析[M]. 北京：电子工业出版社，2011:59-70.

[5] 丰生强. Android 软件安全与逆向分析[M]. 北京：人民邮电出版社，2013:78-90.

[6] 余成锋，李代平，毛永华. Android3.0 内存管理机制分析[M]. 计算机应用与软件

|  |
| --- |
| **班级： 学号： 姓名： （请手写签名）**  **回答问题：** |

|  |
| --- |
| **评语：**  **成绩**  **指导教师**  **年 月 日** |