

# 基于微信小程序的健康饮食软件设计与实现

刘宇豪, 刘佳\*

(武汉商学院信息工程学院, 武汉 430056)

**摘要:** 从开发者的角度以一款健康饮食微信小程序为实例, 简单介绍了其开发过程, 并对其主要功能、软件框架、配置与测试进行了简要阐述。该程序有饮食推荐、食物营养介绍、人体需求、互动平台 4 项主要功能。在阐述完小程序申请开发流程后, 以程序中的全局配置文件 app.json、体质测试模块的 text.wxml、text.js 文件的部分代码, 来说明开发中各属性代码的具体效果。

**关键词:** 健康饮食; 微信小程序; Json

DOI:10.16184/j.cnki.comprg.2020.07.024

## 1 概述

团队曾进行过一次线上问卷调查, 问卷调查有效填写人数为 109 人, 其中 54.13% 认为自己每天的饮食搭配不均衡, 64.22% 认为自己不清楚食物中所含的营养成分, 而 75.23% 并没有用过推荐食谱类的小程序或 APP, 60.55% 认为自己未来会用推荐食谱类的小程序或 APP。因此本项目团队决定制作一款健康饮食小程序, 并推广借此推广楚菜文化。

## 2 小程序主要功能

如图 1 所示。

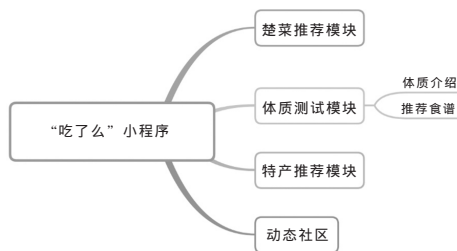


图 1 小程序功能框架图

### 2.1 饮食推荐

用户可以通过小程序进行体质测试, 检测完用户体质后, 程序会根据用户测试出来的体质推荐适合用户、符合用户需求的搭配, 并详细列出用户体质需要注意的健康问题、正确的调理方式及减肥计划等信息, 节省用户大量的学习及规划时间。

### 2.2 食物营养

该模块为学习食物营养的模块, 列举出生活中常见的食物的营养成分、热量、食物属性等信息, 清晰易懂, 解决人们不了解食物营养的问题

### 2.3 人体需求

该模块可通过直观的每日所需营养、水分, 帮助用

户满足自身体质每日所需的营养, 均衡搭配。

### 2.4 互动平台

该项目推出一个交流平台, 可以面对面扫码、输入好友账号或手机号添加好友。同时也可以同步分享到各大交友社交平台, 分享自己的生活, 交流心得体会。

## 3 开发流程

### 3.1 开发环境

微信小程序的开发在本地进行。开发完成后微信审查通过即可发表。

微信小程序的开发, 可以选择微信官方微信开发工具。微信小程序代码编写可参考微信提供的官方开发文档。该文档包含小程序开发的简单教程、框架介绍、组件系统、API 接口、工具等<sup>[1]</sup>。

### 3.2 微信开发者工具

微信开发者工具可实现本地文件同步、开发调试、编译预览、上传、发布等一整套流程。同时提供了很多原生组件, 在 HTML5 中需要模拟才能实现的功能, 在小程序中直接调用组件就可以实现<sup>[2]</sup>。

### 3.3 小程序框架

微信公众平台官网可下载微信开发者工具并申请 AppID, 使用微信开发者工具创建项目时需输入 AppID 或使用测试号。小程序的基本架构是用户与微信进行交互, 微信与 OS 进行交互, 微信通过 API 的方式来获取

基金项目: 武汉商学院 2019 年度大学生创新创业训练项目《吃了么》的研究, 项目编号为 201911654127。

作者简介: 刘宇豪 (2000-), 男, 研究方向: 计算机应用; 刘佳 (1986-), 男, 通讯作者, 教师, 研究方向: 计算机网络、嵌入式系统设计与编程。



第三方的数据，并且通过服务组件、开发工具和交互框架来定义小程序的交互样式、展现方式和扩展功能。如图2，图3所示。

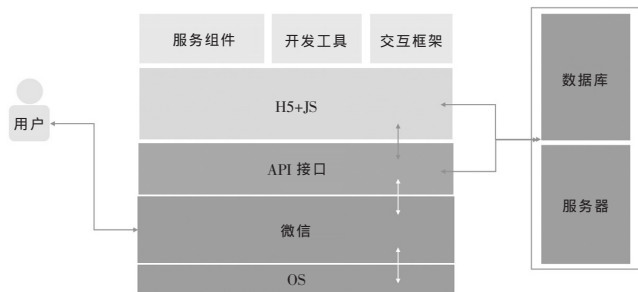


图2 小程序基本框架

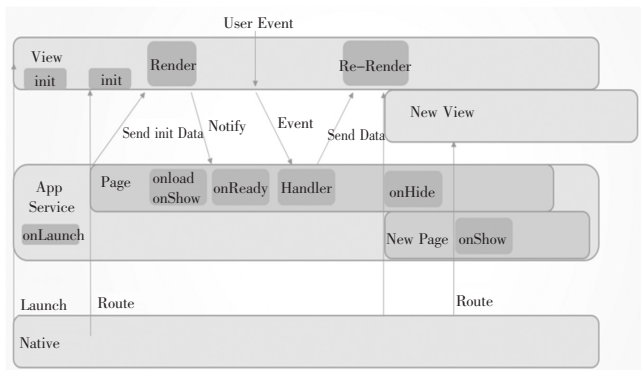


图3 小程序的运行生命周期

## 4 开发实例

### 4.1 app.json 全局配置文件

app.json 是小程序的全局配置，各个页面需在此声明才能使用，页面的底部标签与顶部标签也在此文件中设置。

app.json 文件的声明：

```
{
  "pages": [ //以下小程序各页面的声明
    "pages/subject/subject", //专题页面
    "pages/index_text/index", //体质测试页面
    "pages/test/test", //试题页面
    "pages/result/result", //结果页面
    "pages/techan/techan"],
  "window": { //专题界面顶部设置
    "backgroundTextStyle": "light",
    "navigationBarBackgroundColor": "#494949",
    "navigationBarTitleText": "专题",
    "navigationBarTextStyle": "white",
  },
  "tabBar": { //设置底部表现
    "backgroundColor": "#303133",
    "color": "#ffffff",
  }
}
```

```
"selectedColor": "#CC1004",
"list": [
  {
    "pagePath": "pages/subject/subject",
    "text": "专题",
    "iconPath": "pages/images/tab/subject-0.jpg",
    "selectedIconPath": "pages/images/tab/subject-1.jpg"
  }
],
"sitemapLocation": "sitemap.json"
}
```

在该文件中"pages": [] 用于存放页面路径，"window": {} 属性设置全局的默认窗口表示"tabBar": {} 属性设置底部tab栏的表现，在此程序中，底部tab栏以“专题”模块为例，由图像引导跳转，用户所在进入模块后的tab图像与进入前的不同，以此区别模块状态并提示用户目前进行浏览的模块的位置。

### 4.2 测试模块

测试模块根据用户所做的测试结果告知用户自身的体质，题库储存于app.js文件中，答题页面为text.wxml文件，答题模块的逻辑储存于text.js文件中，用户答题结束，result页面会根据用户所选的选项进行判断，显示用户的体质类型。

text.wxml 文件内容：

```
<view class='page_hd'>
  <view class="page_title">{{index+1}}、{{question-
Detail}}</view>
</view>
<view class="weui-cells weui-cells_after-title">
  <view url="" class="weui-cell weui-cell_access"
  hover-class="weui-cell_active" bindtap='answerClickA'>
    <view class="weui-cell_bd">{{optionA}}、
    {{answerA}}</view>
  </view>
  <view url="" class="weui-cell weui-cell_access"
  hover-class="weui-cell_active" bindtap='answerClickB'>
    <view class="weui-cell_bd">{{optionB}}、
    {{answerB}}</view>
  </view>
  <view url="" class="weui-cell weui-cell_access"
  hover-class="weui-cell_active" bindtap='answerClickC'>
    <view class="weui-cell_bd">{{optionC}}、
    {{answerC}}</view>
  </view>
</view>
```

&lt;/view&gt;

text.js 文件部分内容:

```
// pages/test/test.js
//获取应用实例
const app = getApp()
Page({
  // 页面的初始数据
  data: {index: 0,realIndex: 0,A: 0,B: 0,C: 0,a:0,b:0,c:0,
    optionA: "A",optionB: "B",optionC: "C",
    questionDetail: app.globalData.question[0].question,
    answerA: app.globalData.question[0].option.A,
    answerB: app.globalData.question[0].option.B,
    answerC: app.globalData.question[0].option.C,
    list: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
    15, 16, 17, 18, 19],
    listABC: ['A','B','C']},
  randSort: function() {return Math.random() > 0.5 ? 1
: -1;},
  setList: function () {var newList = this.data.list.sort
(this.randSort);
  this.setData({list: newList,});},
  setABC : function(){var abc = this.data.listABC.sort
(this.randSort);
  this.setData({listABC: abc,});},
  setOption: function(),
  onShareAppMessage: function() {}})
```

上述 text.js 文件省略了判断答案的具体步骤, 其流程为判断用户选择各答案的数量, 不同数量的答案对应不同的体质结果, 答案与体质的对应关系不涉及网站的技术问题, 故在此忽略其实现代码。

在 text.wxml 文件中, <view class=" page\_title" > {{index+1}}、 {{questionDetail}} </view>为题干, 题目的选项由 A、B、C 3 项, 介绍代码以选项 A 为例: <view url="" class=" weui-cell weui-cell\_access" hover-

(上接第 34 页)

层都支持横向扩容, 可以很容易地从两台服务器扩充到数百台服务器, 按照单台 200qps, 每个用户每 5 秒点击 1 次计算, 可以轻松应对数十万用户同时使用。同时并不会增加系统开发难度, 对于程序员来说几乎是透明的。

系统的高可用性并非只有一种解决方案, 以上是对简单系统的一种常见实践。随着用户量增加, 系统功能变得复杂, 系统开发人员会面临更多的问题, 比如将一个系统按照业务逻辑拆分成多个系统后, 系统和系统之

class=" weui-cell\_active" bindtap='answerClickA'><view class=" weui-cell\_bd" > {{optionA}}、 {{answerA}} </view></view>为 A 选项, 点击选项后触发 answerClickA 函数使变量 A+1, 完成测试后在跳转到 result 页面, 并统计用户选择各选项的次数, 最后根据选项结果判断用户体质。如图 4, 图 5 所示。



图 4 测试页面



图 5 主页页面

## 5 结语

微信“小程序”与应用程序相比系统架构更简单。同时, 微信提供了必要的开发人员工具, 使开发人员能够轻松地开始开发微信“小程序”, 对开发的技术要求更低, 并通过微信自带的流量使程序的宣发更加容易, 适合资金较少团队进行开发。

### 参考文献

- [1] 王玉洋. 基于微信小程序的移动学习平台环境构建与系统设计开发 [D]. 南京: 南京大学, 2018.
- [2] 武思怡, 解佳, 张雨, 侯秉昀. 微信小程序开发研究 [J]. 无线互联科技, 2018, 15 (11): 52-54.

间如何相互调用, 如何保证服务终止后能及时发现并主动调用其他同类型的服务; 以上实现中的数据层采用了分片的方式, 虽然把鸡蛋分开放在了几个篮子里, 但是每次宕机依旧会影响部分用户, 是否可以采用共享磁盘的方式继续减少风险。当然采用以上设计后, 硬件成本也大大上升, 从原来的一台服务器, 变成了至少 7 台服务器, 所以要根据自身的实际情况采用最合理的高可用解决方案。

