

# 第二组 小程序介绍文档

组长：刁苏阳 181250026

组员：陈思文 181250013；李宇轩 181250076；

高金比尔 181250035；戴祺佳 181250023；

## 概要

名称：NJUSE 饮食小管家

项目背景：随着现代人生活条件逐步提高，饮食、健身等观念也走入大众的视野。饮食小管家这一个项目从大众健康饮食的角度出发，提供机器人交互的趣味形式，方便群众合理安排并记录自己的日常饮食，并提供目标、成就等鼓励措施引导群众自我监督、自我约束。

目标对象：具有饮食管理需求的用户

功能概述：提供饮食记录、健康饮食推荐等服务功能，方便用户进行自身健康饮食管理

## 功能点

### 1. 机器人（基于微信对话开放平台实现，手机版支持语音输入）

#### 1.1 微信对话开放平台：平台提供功能：聊天、天气查询等（左图一）



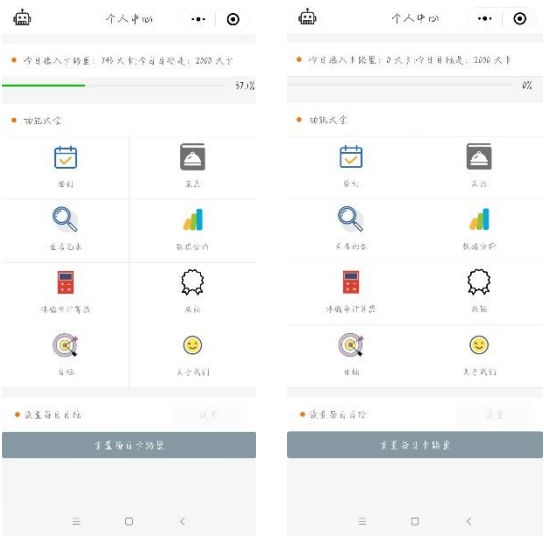
1.2 扩展平台功能：帮助指引、食谱推荐（左图 2、3）

1.3 平台插件重载：饮食记录（和每日饮食目标相关联）（右图 1）

2 每日饮食

根据平台插件重载调用用户端信息，并根据用户信息修改每日摄入的卡路里数量，支持用户进行重置和目标设定等功能

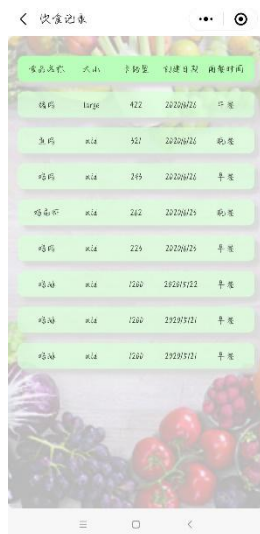
（下图 1、2）



3 菜品搜索和添加（下图分别展示两个界面）



## 4 饮食记录查看



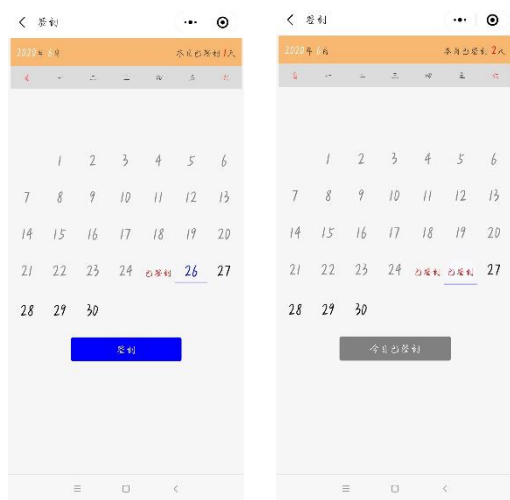
## 5 每餐饮食数据分析（统计卡路里最多的食物和最少的食物以及平均摄入



## 6 体脂率计算（数据输入后会计算出对象体脂率



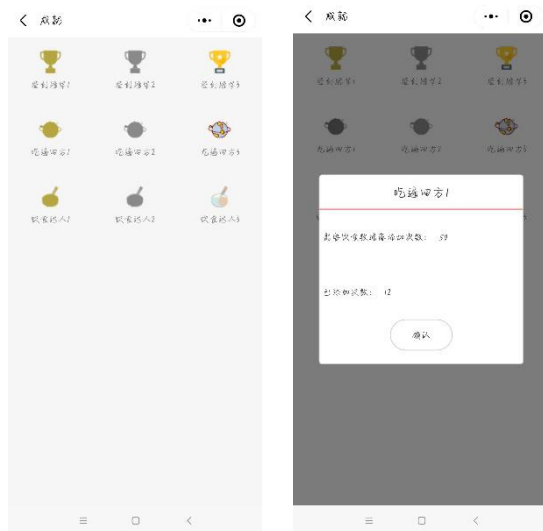
## 7 签到（展现签到前后对比）



## 8 目标（于上方输入栏添加目标至未完成中，点击未完成目标栏中的“完成”按钮修改目标状态为已完成，实现自我监督



## 9 成就（图形界面连接数据库，可以查看用户成就方面活动信息



## 使用技术介绍

代码开源地址: <https://git.weixin.qq.com/181250026/weChatProgram.git>

### 1. 机器人

#### 1.1 微信对话开发平台机器人调制:

详情: 微信对话开放平台机器人申请及创建, 自定义对话功能, 机器人调试及线上发布

#### 1.2 插件接入:

详情: 程序内引入插件 **openid** 以及版本号, 在页面初始化后设置并调用

#### 1.3 插件重载:

详情: 插件页面引入插件重载相关 **component** 对象, 重写插件的用户信息读入和服务端信息输出组件, 可凭用户输入特定信息对指定数据库进行增删改查

#### 1.4 代码示例: 展示插件初始化以及插件重载部分

```

plugin.init({
  appid: "8jI98IX8iynnb9fgaE2IXvRowsSp4",
  openid: "wxe864426fbf8e9b5",
  guideList: ['/帮助', '/健康食谱推荐', '健身吃什么'],
  welcome: '小伙伴, 你要找我小智随玩什么呢?',
  background: "#eee",
  textToSpeech: false,
  guideCardHeight: 50,
  operateCardHeight: 0,
  history: false,
  historySize: 60,
  navHeight: this.data.statusBarHeight,
  robotHeader:
    "https://res.wx.qq.com/mmspraiweb_node/dist/static/openaipugin/img/answerImage.png",
  success: () => {
    this.setData({
      flag: true
    })
  },
  fail: error => {}
});

```

```

methods: {
  choose: function(e) {
    const chat = plugin.getChatComponent();
    chat.send(e.currentTarget.dataset.title);
  },
  getFacePosition(index) {
    const row = 15;
    const width = 24;
    const height = 24;
    const y = Math.floor(index / row);
    const x = index - y * row;
    return -(x * width) + "px " + -(y * height) + "px";
  },
  tap: function(e) {
    let url = e.currentTarget.dataset.weburl;
    wx.navigateTo({
      url: `./pages/webviewPage/webviewPage?url=${url}`
    })
  },
}

```

## 2. 数据库：

### 2.1 云数据库

微信云开发提供云 JSON 数据库，支持回调风格与 Promise 风格，小程序各部分调用云环境对云数据库进行增删改查操作

提供数据库读写权限配置，可让不同功能的数据集合具有不同的用户读写权限

### 2.2 云函数调用

回调函数操作和异步处理

### 2.3 代码展示：数据库访问

```

/**
 * 获取某用户的所有记录
 * @param {指向发起调用的页面} that
 * @param {调用者页面的名称} pageName
 */
function getAllRecord(that, pageName="showRecordList") {
  const db = wx.cloud.database({
    env: 'weixin-cloud-g1o7d'
  })
  var userID
  wx.getUserInfo({
    success: res => {
      userID = res.signature
    }
  })
  db.collection('record').get({
    success: res => {
      wx.showToast({
        title: '拉取成功',
      })
      that.setData({
        recordList: res.data.reverse()
      })
    }
  })
}

```

### 3. 图形界面:

#### 3.1 WE UI:

WeUI 是一套同微信原生视觉体验一致的基础样式库，为微信 Web 开发量身设计，可以令用户的使用感知更加统一。包含 **button**、**cell**、**dialog**、**toast**、**article**、**icon** 等各式元素。我们本次作业中的按钮、文本框等元素用到了 WEUI 的属性

#### 3.2 CSS 渲染:

CSS 可以控制 HTML 标签对象的 **css 宽度**、**css 高度**、**float 浮动**、**文字大小**、**字体**、**css 背景**等样式达到我们想要的布局效果。由于 **css** 的渲染规则，是从上到下，从右到左渲染的，所以我们本次作业的嵌套层级没有超过三级，以防止过度的嵌套导致代码变得臃肿，导致 **css** 文件体积变大，造成性能上的浪费，影响渲染的速度；**css** 层级太多，也过于依赖 **html DOM** 结构，不易于维护。如果层级比较深，我们就直接定义一个 **class** 代替多余的层级。并且尽可能的继承父类样式，重复的定义会造成很多不必要的性能浪费。

#### 3.3 代码示例:

```
height: 60px; /*高度*/
display: flex;
line-height: 40px; /*内框高度*/
flex-direction: row; /*排列方式*/
text-align: center; /*文字格式*/
background-color: #1851850.75; /*背景颜色、透明度*/
border-top-left-radius: 10px;
border-top-right-radius: 10px; /*边框*/
padding: 10px;
box-shadow: 10px 10px 5px #1851850.75; /*阴影*/
margin-left: 15px; /*左间距*/
margin-right: 15px; /*右间距*/
border: 1px solid #1851850.75;
z-index: 1; /*图层*/
position: relative;
}
.background {
width: 100%;
height: 100%;
background-size: 100% 100%;
position: fixed;
z-index: -1;
box-shadow: 10px 10px 5px #1851850.75;
border-radius: 10px;
} /*背景*/
```

```
<view>
  /*背景图片*/
  <image class="background" src="../../images/health.jpg" style="opacity:{{0.25}}" mode="aspectFill"></image>
  <view class="cu-form-group margin-top" style="margin-left: 15px; margin-right: 15px; border-radius: 10px; background-color: opacity:{{0.25}}">
    <input bindinput="successInput" placeholder="添加目标" value="{{name}}"></input>
    <button class="cu-btn bg-grey lg" hover-class="none" bindtap="getvalue" disabled="{{disabled}}" style="width: 30vw">添加</button>
  </view>
  /*未完成的目标*/
  <view class="content1">
    <view style="flex: 1; height: 20px">未完成</view>
  </view>
  <view class="content2" wx:for="{{failList}}" wx:key="name" data-id="{{item._id}}">
    <view style="flex: 1; height: 20px;">{{item.name}}</view>
    <button class="cu-btn bg-grey lg" hover-class="none" bindtap="setFinish" style="width: 20vw" data-id="{{item._id}}">完成</button>
  </view>
  /*已完成的目标*/
  <view class="content1">
    <view style="flex: 1; height: 20px">已完成</view>
  </view>
  <view class="content2" wx:for="{{successList}}">
    <view style="flex: 1; height: 20px;">{{item.name}}</view>
  </view>
</view>
```

# 成员分工介绍

刁苏阳：机器人调制、接入、重载；成就页面数据库接入；每日饮食接口对接

陈思文：整体布局设计、图形界面渲染；关于我们 页面实现

李宇轩：页面跳转、体脂率计算 功能实现；成就界面页面渲染

高金比尔：签到、菜品、目标 功能实现；云数据库、云函数访问

戴祺佳：查看记录、数据分析、每日饮食功能实现；云数据库、云函数访问