



南開大學  
Nankai University

《智能移动开发》课程大作业  
“乡村志愿汇”小程序项目介绍

1913183

孙宜辰

2022 年 6 月 10 日

# 乡村志愿汇

## 1 需求功能说明

### 1.1 需求背景

农村志愿服务者在扶助困难群体、发展农业生产、促进邻里互助、倡导文明风尚、关怀儿童教育、协调矛盾冲突、保护生态环境等方面成为农村社会和谐稳定不可忽视的力量。但同时，由于乡村渠道受限、信息不畅，志愿者招募方式单一、范围小、质量低；受困管理机制的缺陷，志愿者积极性和持续性差。希望可以利用互联网工具，构建一个以政府为主导，机构为发起者，媒体为媒介，社区、高校、企业为响应者的新机制，重视招募的规范性并设立相应门槛，在管理层面完善机制。本项目选取 2022 届微信小程序开发赛“乡村志愿者征集工具”主题进行了开发。

### 1.2 产品功能

#### 用户页

- 用户登录及退出
- 查看志愿时长，活动时长，志愿积分
- 查看已报名的活动，已加入的组织
- 查看并修改个人资料
- 咨询客服

#### 1.签到页

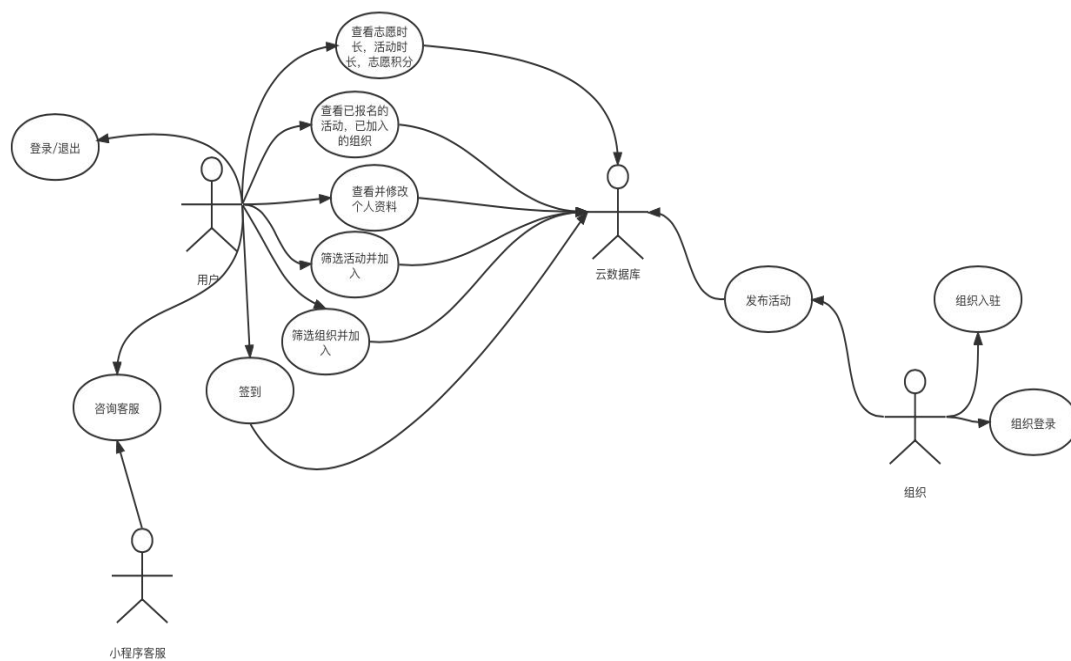
- 进行对当日正在进行已报名活动的签到

#### 2.首页

- 在首页显示正在招募的最新活动
- 按关键词和地点筛选组织并加入
- 按关键词和地点以及活动类型筛选志愿活动并加入

- 组织注册提交审核
- 以组织身份登陆并发布活动

### 1.3 用例图

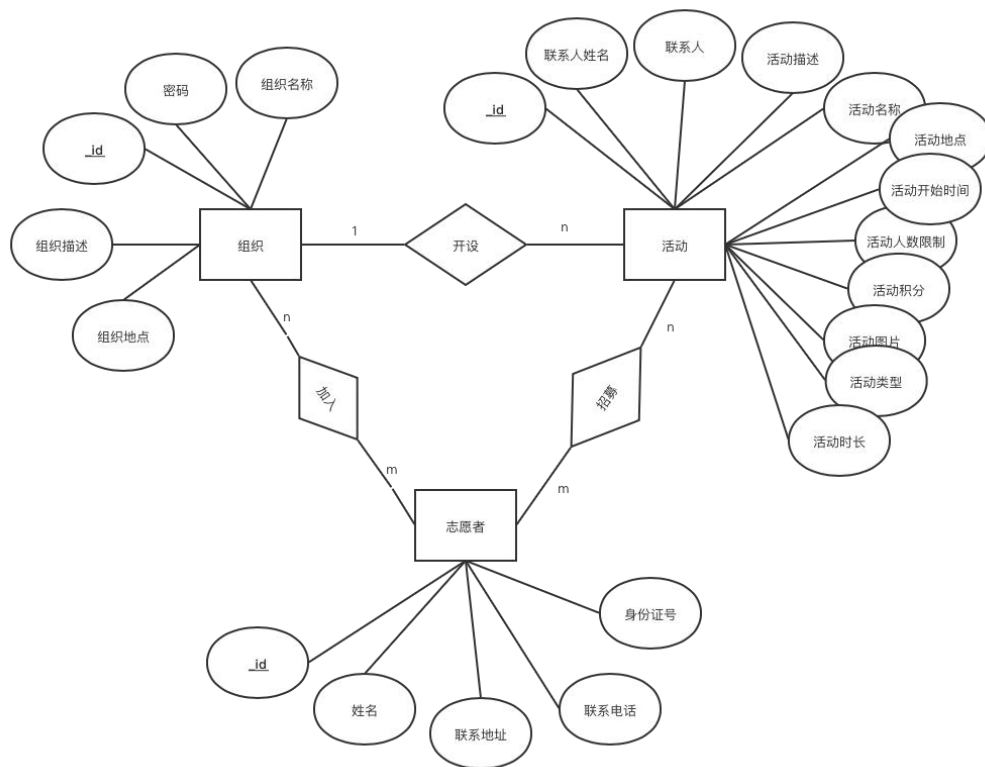


## 2 需求功能说明

### 2.1 技术选型

本项目采用微信团队联合腾讯云提供的原生 **Serverless** 云服务，未搭建后端，仅采用平台提供的各项能力：云数据库，云存储，云函数，云调用。

### 2.2 云数据库设计



## 2.3 数据来源

从“志愿汇”程序获取活动和组织等信息，存储在云数据库中。

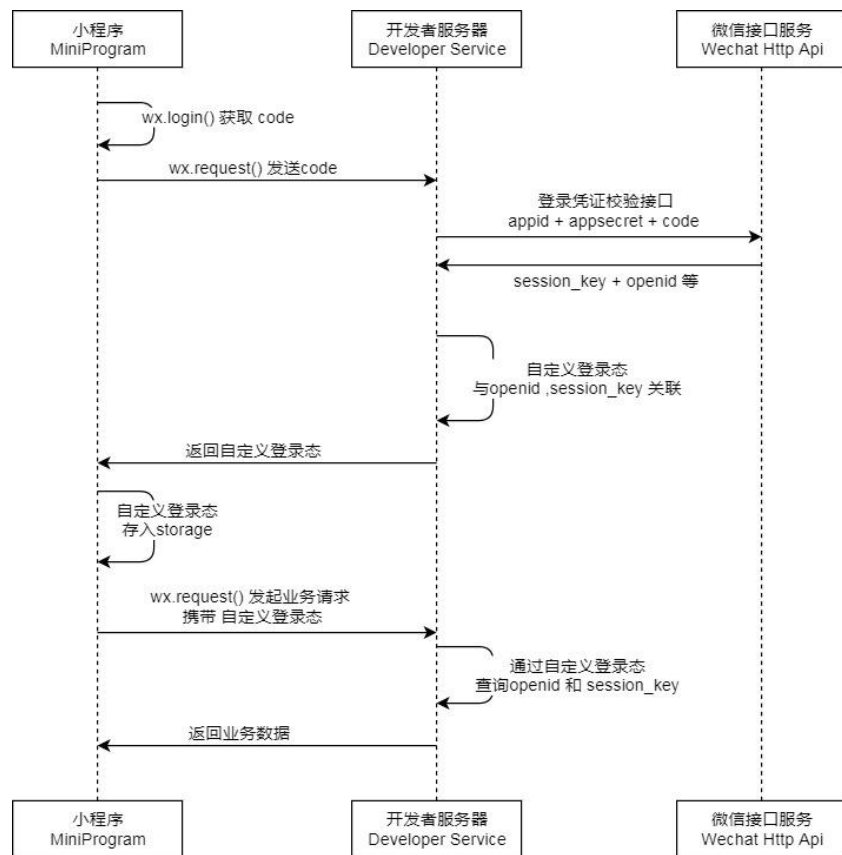
## 3 产品实现说明

### 3.1 用户页



#### 3.1.1 登录获取用户信息

登录通过调用 `wx.login` 接口获取登录凭证（`code`）。通过凭证进而换取用户登录态信息，包括用户在当前小程序的唯一标识（`openid`）、微信开放平台帐号下的唯一标识（`unionid`，若当前小程序已绑定到微信开放平台帐号）及本次登录的会话密钥（`session_key`）等。用户数据的加解密通讯需要依赖会话密钥完成。

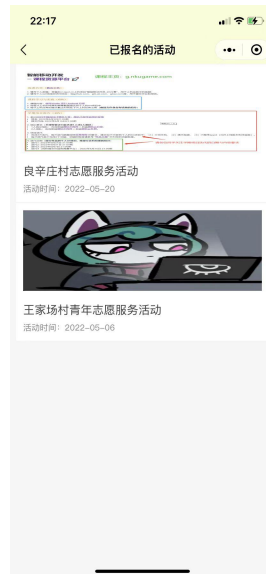


获取用户信息通过调用 `wx.getUserInfo` 接口实现，该接口只能在获得用户授权后调用，为了更加便捷的获取敏感数据，采用云调用直接获取。

### 3.1.2 查看志愿时长、活动次数、志愿积分

登录进小程序后首先获取用户的 `openid`，用于唯一标识用户，然后从云数据库的 `User` 表中使用 `Collection.get()`请求获取用户的相应信息。

### 3.1.3 查看已报名的活动、已加入的组织



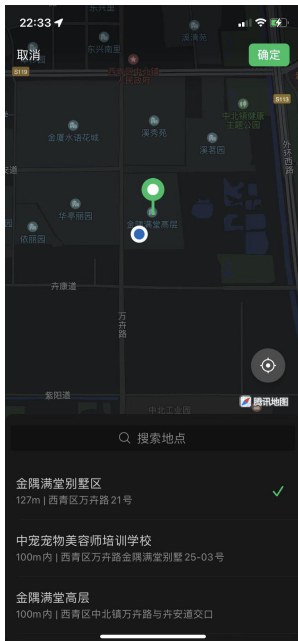
在用户页从云数据库的 `User` 表请求获取到用户的活动信息和组织信息后，使用 `wx.navigateTo` 保留当前页面，跳转到已报名活动/组织页面，同时将对应信息传值到新页面中。

### 3.1.4 查看修改个人资料



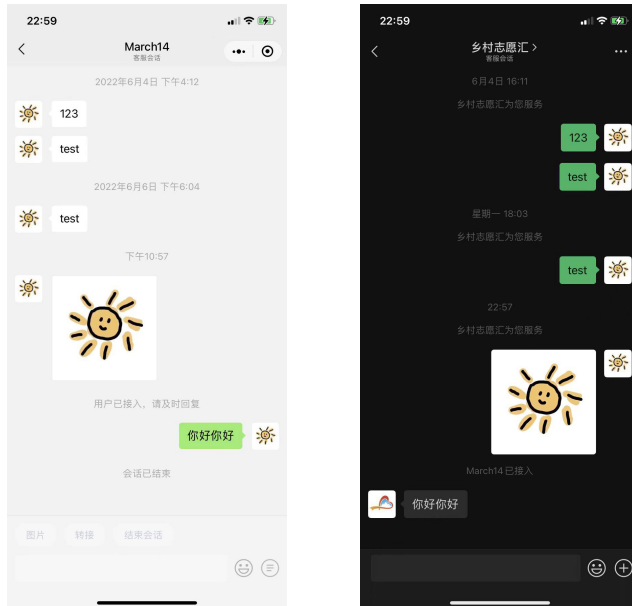
在该页面可进行对默认个人资料的查看并修改提交

在用户页从云数据库的 `User` 表请求获取到用户信息后，通过 `wx.navigateTo` 跳转到本页，在本页的 `wxml` 文件中使用一个表单组件。其中，点击“联系地址”后会弹出地图，这是通过 `wx.chooseLocation` 接口实现的。



在点击提交按钮后，程序将调用 `collection.update` 请求更新云数据库 User 表中对应的个人信息

### 3.1.5 联系客服



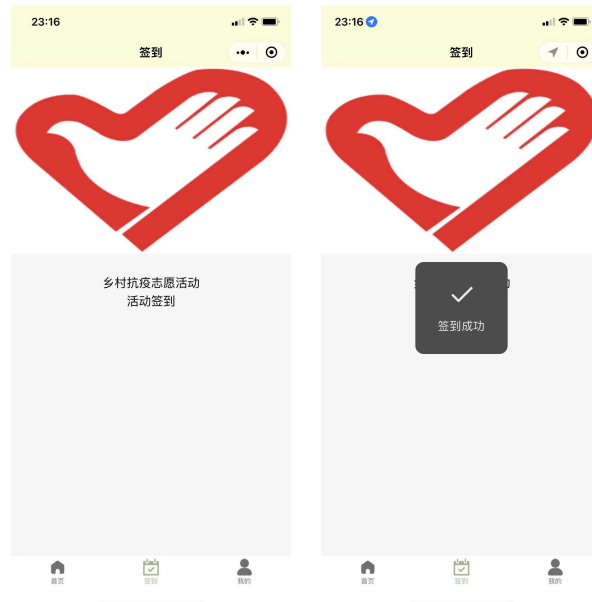
在该页面的 wxml 文件中通过将按钮的 `open-type` 设置为 `contact` 实现

```
<view class="list">
  <button open-type="contact">联系客服</button>
  <image class="jiantou" src="../../images/user/youjiantou.png"></image>
</view>
```

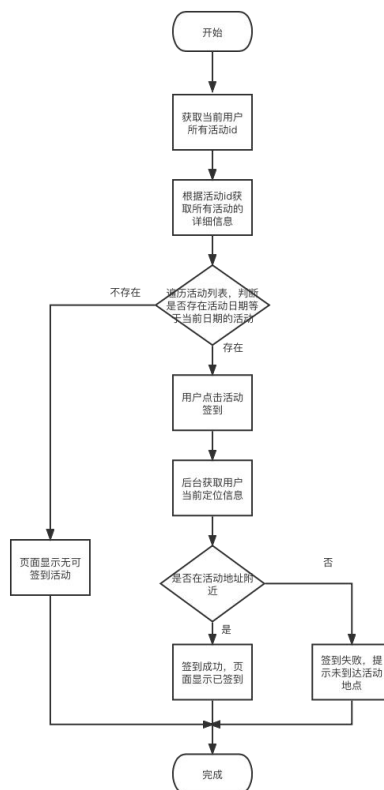


## 3.2 签到页

### 3.2.1 对当日正在进行的活动进行签到



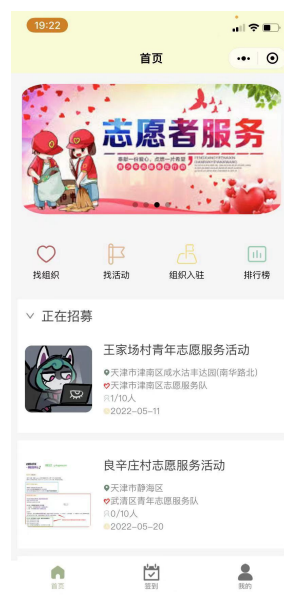
该功能的实现的流程图如下图所示：



获取用户的后台定位功能本通过微信平台的地址解析 api 实现，应在 app.json 中开启相应的接口权限。

```
wx.getLocation({
  success: (res) => {
    var latitude=this.data.detail[0].location.latitude
    var longitude=this.data.detail[0].location.longitude
    console.log(latitude,longitude)
    console.log(res)
    if(res.latitude<latitude+0.1){
      console.log("签到成功")
      wx.hideLoading()
      wx.showToast({
        title: '签到成功',
        duration:1000
      })
      this.setData({
        flag:true,
      })
    }else{
      wx.hideLoading()
      wx.showToast({
        title: '未到达活动地点',
        icon:"error",
        duration:1000
      })
    }
  }
})
```

### 3.3 首页



首页加载时会在 `onload` 中调用 `collection.get` 请求从云数据库中获取活动列表，按时间排序后展示在首页

### 3.3.1 筛选活动及活动报名



从云数据库中的 `activities` 表中获取活动数据，用户可按照活动类型和关键字以及地点进行筛选。

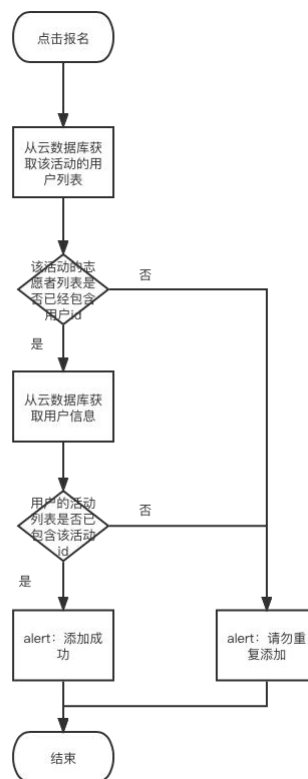
该页面的 `wxml` 文件中设置了两个 `picker`，对应活动类型和活动所在区域，用户在改变 `picker` 的值时会调用 `bindRegionChange` 和 `bindTypeChange` 函数，改变筛选活动的条件，从而改变云数据库查询操作的参数。关键字筛选功能则直接在活动列表中查找包含关键字的活动信息。

当用户点击活动时，会跳转到活动详情页，并使用 `wx.navigateTo` 向活动详情页传入活动信息。



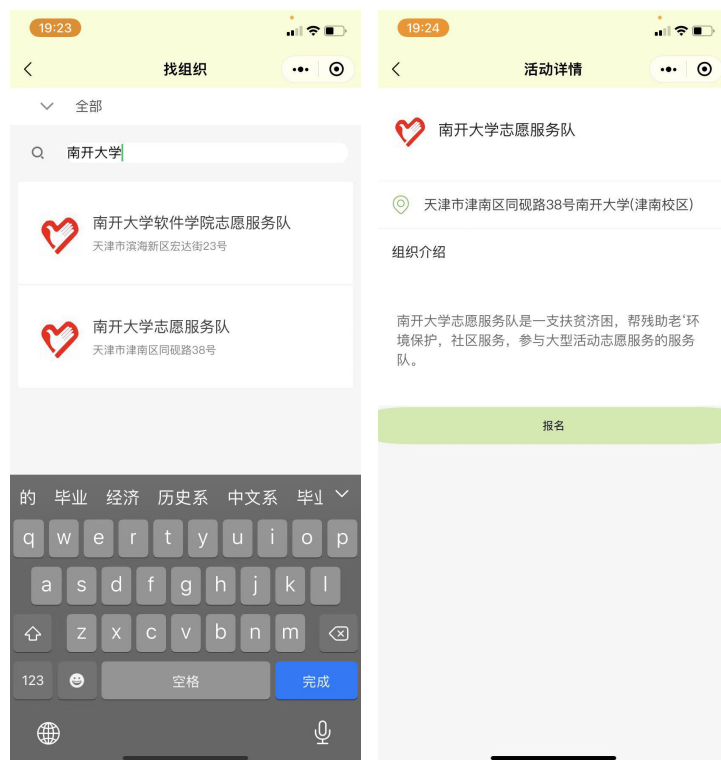
加入活动，首先从云数据库获取活动的详细信息，判断活动的报名成员中是否包含用户，若不包含，则判断用户是否已报名该活动，若不包含，则在用户的活动列表中添加该活动 id，在活动的报名表中添加用户 id，报名成功。否则显示重复添加。

流程图如下所示：



### 3.3.2 筛选组织并加入

从云数据库中的 **organization** 表中获取活动数据，用户可按照关键字和地点进行筛选。

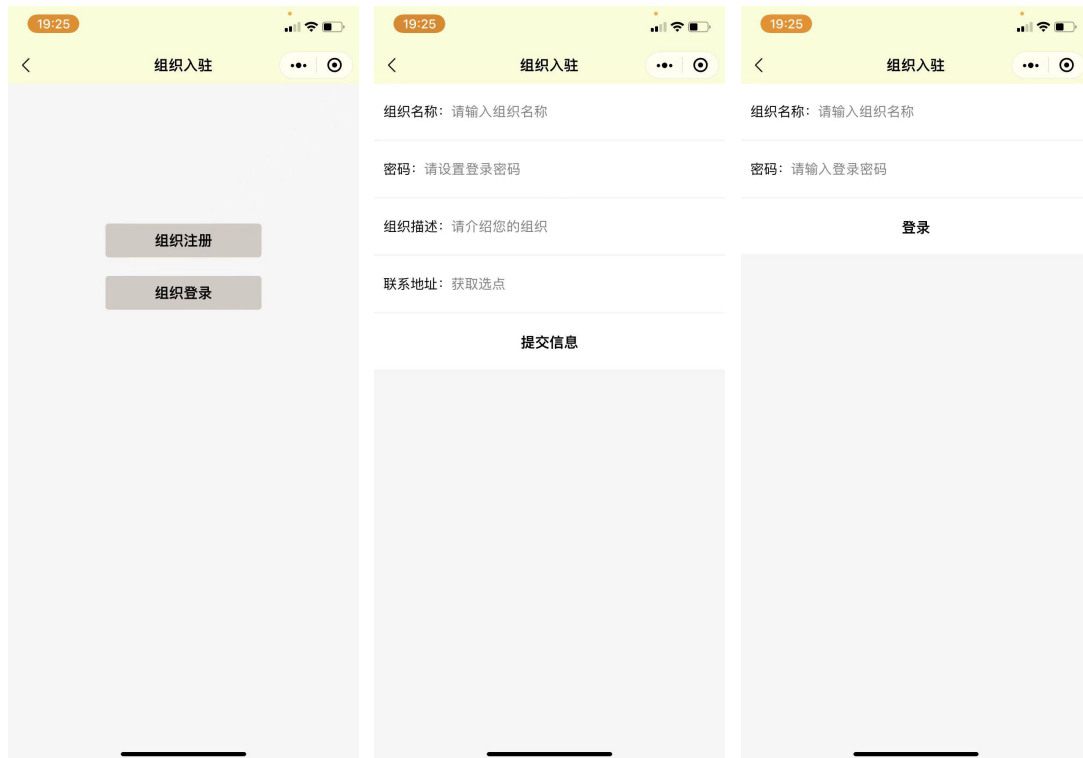


该页面的 **wxml** 文件中设置了一个 **picker**，对应组织所在区域，用户在改变 **picker** 的值时会调用 **bindRegionChange** 函数，改变筛选组织的条件，从而改变云数据库查询操作的参数。关键字筛选功能则直接在组织列表中查找包含关键字的活动信息。

当用户点击组织时，会跳转到组织详情页，并使用 **wx.navigateTo** 向组织详情页传入组织信息。

加入活动，首先从云数据库获取组织的详细信息，判断组织的成员中是否包含用户，若不包含，则判断用户是否已加入该组织，若不包含，则在用户的组织列表中添加该组织 **id**，在组织的报名表中添加用户 **id**，加入成功。否则显示重复添加。

### 3.3.4 组织注册/登录



用户注册页面的 `wxml` 文件使用一个表单组件，当用户提交注册信息后，将调用 `collection.add` 请求向云数据库中的 `Organizaition` 表中添加新的记录。注册后用户将以组织身份自动登录。

用户登录页面点击登录按钮时，将会调用 `collection.get` 请求获取组织名称对应的组织信息并判断密码是否正确。

### 3.3.5 发布活动



用户以组织身份登录后方可以发布活动，发布活动的页面的 `wxml` 文件中使用一个表单组件，可通过 `picker` 选择活动日期，和活动类型。点击上传图片后，会调用云函数 `wx.cloud.upload` 将用户上传的图片存入云存储中，然后将云存储中的文件 `id` 存入云数据库中的 `imgUrl` 字段。

```
upload(){
  wx.chooseImage({
    count: 1,
    sizeType: ['original', 'compressed'],
    sourceType: ['album', 'camera'],
    success: res=>{
      // tempFilePath可以作为img标签的src属性显示图片
      const tempFilePaths = res.tempFilePaths
      console.log("xuanzechenggong", res)
      wx.cloud.uploadFile({
        cloudPath: new Date().getTime()+'.png', // 上传至云端的路径
        filePath: tempFilePaths[0], // 小程序临时文件路径
        success: res => {
          // 返回文件 ID
          console.log('shangchuanchenggong', res.fileID)
          this.setData({
            imgUrl: res.fileID
          })
        },
        fail: console.error
      })
    }
  })
}
```