基于微信小程序的

景区导航排队系统

产品设计文档

**产品代号：**

**编写人员：**

**编写时间：2022年4月17日**

**修改日期：2022年4月17日**

目录

[一、 系统说明 1](#_Toc22429)

[二、 实验环境 2](#_Toc18537)

[三、系统需求分析和功能设计 2](#_Toc9725)

[1.系统的需求分析 2](#_Toc28612)

[2.系统功能设计 2](#_Toc21401)

[2.1智能景区导航系统设计 2](#_Toc27278)

[2.2 游客购买门票功能设计 2](#_Toc3198)

[2.3 旅游导航功能设计 3](#_Toc10162)

[2.4登录功能 3](#_Toc29371)

[四、系统设计 4](#_Toc7333)

[功能图 4](#_Toc6917)

[流程图 5](#_Toc5111)

[数据库及表的设计 5](#_Toc21405)

[五、系统实现 7](#_Toc16685)

[5.1小程序云开发项目的初始化 7](#_Toc12072)

[5.2小程序目录结构 8](#_Toc19617)

[5.3小程序框架分析 9](#_Toc23460)

[5.4小程序常用组件 9](#_Toc24388)

[5.5数据绑定 9](#_Toc19866)

[5.6小程序常用API 10](#_Toc13674)

[5.7小程序云数据库 11](#_Toc18997)

[5.8云函数 12](#_Toc14607)

[5.9腾讯地图开放开台 12](#_Toc28412)

# 系统说明

设计并实现一个旅游景区导航排队系统的小程序端。自新冠疫情爆发以来，以人员流动性，聚集性为特征的旅游业遭到了巨大冲击。随着各地对常态化疫情的防控，国内旅游业的恢复形势总体向好，项目可以帮助旅游产业的复苏。

# 实验环境：

Windows 10专业版、微信开发者、微信小程序云开发平台

# 三、系统需求分析和功能设计：

## 1.系统的需求分析

智慧旅游是旅游发展的必然趋势，它将计算机信息技术和旅游业的发展相结合。智能景区导航系统则是通过游客的移动终端机获取到自带的GPS 定位系统中来得到游客的位置信息，并将该信息实时传递给电子地图，且能将该位置标注在电子地图中。景区导览系统采用ESP-CAM作为图像传感器负责采集景点实时图像作为原始数据。ESP-12K数据传输网关通过WIFI获取所在景点的多个ESP-CAM传感器采集的图像，将其通过网路传输到云端服务器。在综合应用层采用小程序与用户就行交互，小程序端接收云端传过来的数据，处理后展示在用户面前，向用户展示景点信息，如名称，地点，开放时间，排队等候人数，预计排队时间。另有购买门票功能，用户可自行选择买票方式。实现自主式游览，以此来满足游客的个性化需求，提高旅游区的服务质量。

## 2.系统功能设计

### 2.1智能景区导航系统设计

智能景区的导航系统主要是让游客以自助的形式来增强游客的互动性和参与性，并提供更加便捷化的旅游景点服务，而当中最重要的就是游览地图的导航，导览地图的设计要 更加立体并提供全方位展示和导航服务，以此来满足游客的多样性和个性化需求。 智能景区导览系统总体架构将智能景区导览系统划分为两个部分，部分之间相互分离，使得系统中的各个功能作用更加明确，能够最大化地降低系统的耦合性，提高系统性能。

### 2.2 游客购买门票功能设计

该功能信息是通过游客在小程序端提交订单，后台服务器读取提交的订单，由景区管理人员在后台数据端进行的买票操作。

### 2.3 旅游导航功能设计

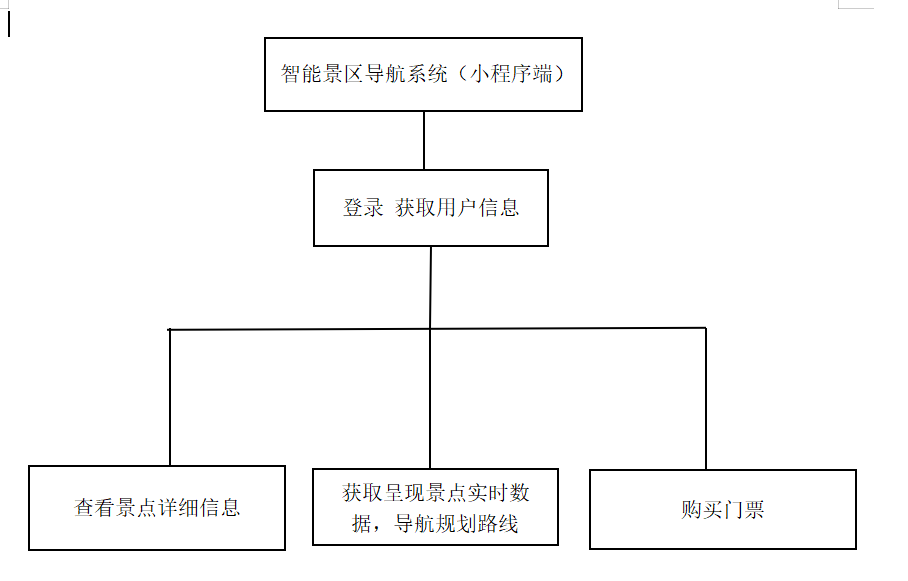
旅游导览功能主要实现用户通过客户端能自动定位、浏览景区的地图、景点的导航等。客户端自动定位位置首先需要客户端终端机连接网络，然后根据终端机自带的 GPS定位功能将游客的位置信息显示在地图中，定位到自身的位置后能够计算当前到最近景点的距离以及附近景点的推荐和介绍，并在游客选中目的地后就能直接进入导航界面。景点信息存储在云数据库中，同时接收后端服务器实时传输的动态数据，如当前景点人数。景区地图和导航功能主要是对整个景区的介绍和主要路线的筛选，客户端能够将整体的景区地图及当前景点的数据下载到游客终端机中，游客就能够根据自身的需求来选择路线或规划路线，以及进行路线的引导。

### 2.4登录功能

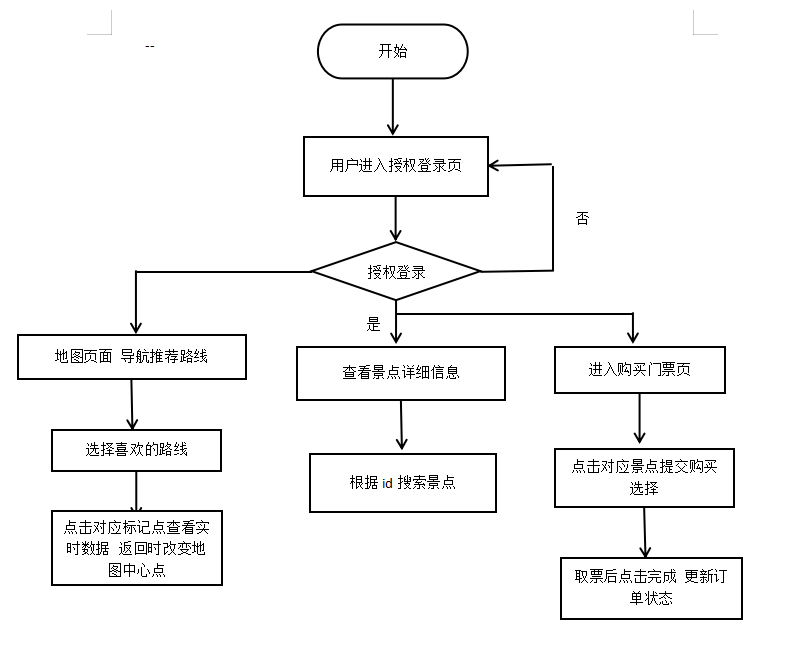
该功能主要实现游客数据的采集，通过游客点击授权来获取信息，然后将其保存到数据库中进行分析处理。此处获取的用户信息将用于用户买票及查看已购门票信息。

# 四、系统设计（包括功能图、流程图、数据库及表的设计）：

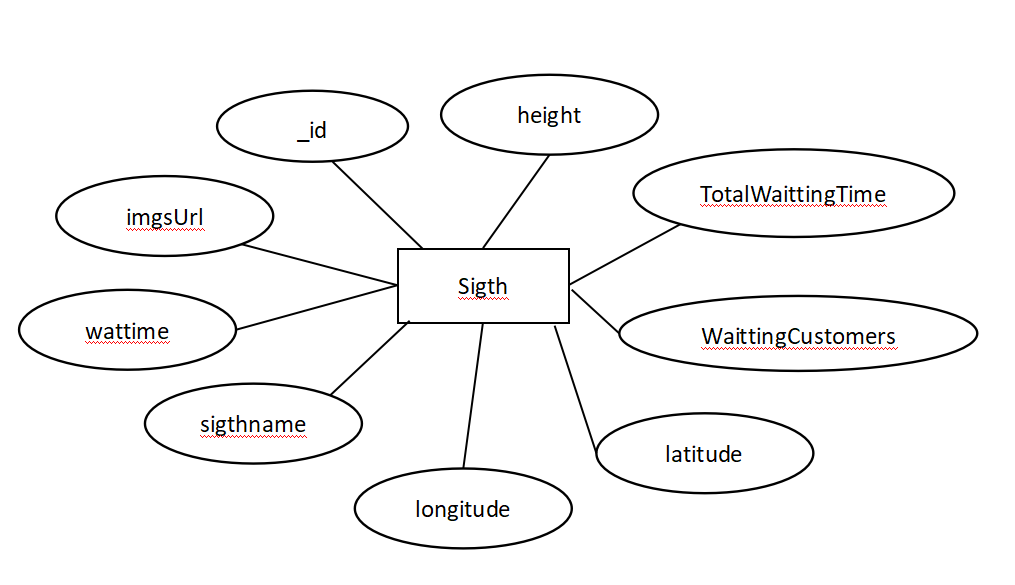
## 功能图：



## 流程图：

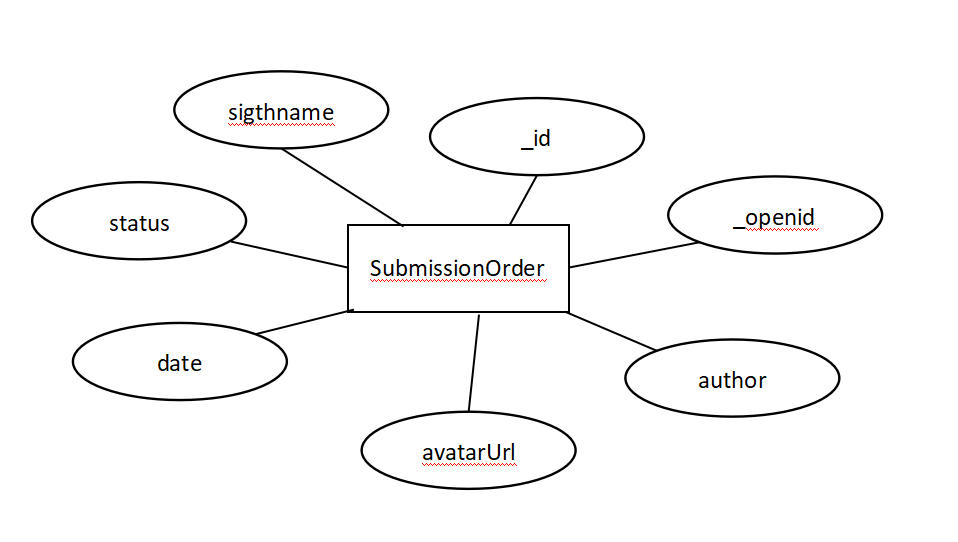


### 数据库及表的设计：



1

提交

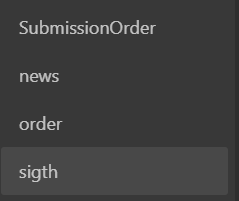


n

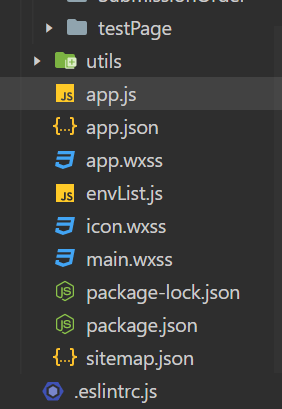
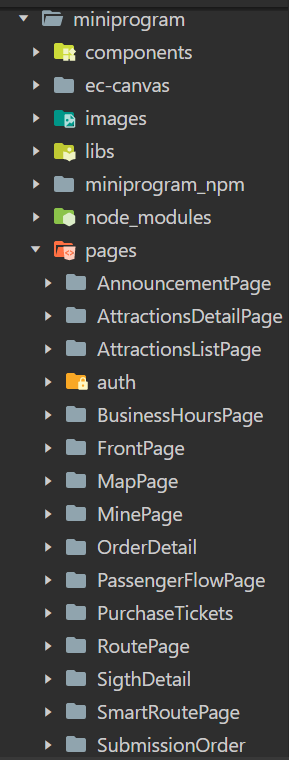
# 五、系统实现：

### 5.1小程序云开发项目的初始化：





### 5.2小程序目录结构：



### 5.3小程序框架分析：

逻辑层：js文件用来处理业务逻辑。

视图层：wxml文件用来渲染页面，wxss文件用来设置样式。

### 5.4小程序常用组件：ColorUI组件、We-Ui组件库

常用colorui组件的头像、表单组件，weui的半屏弹窗组件

### 5.5数据绑定

从云数据库获取数据，在js文件中用setdata方法绑定，在wxml页面用{{}}渲染，将云数据库中数据显示到页面上。

db.collection('sigth').where({

id:sigthId

}).get().then(res=>{

var sigth=res.data[0];

// sigth.data=new Date(sigth.data).toLocaleDateString();

this.setData({

sigth:sigth

})



### 5.6小程序常用API

获取用户信息：

wx.getUserProfile({

desc: '获取用户信息',

success:(res)=>{

console.log(res);

app.globalData.userInfo=res.userInfo;

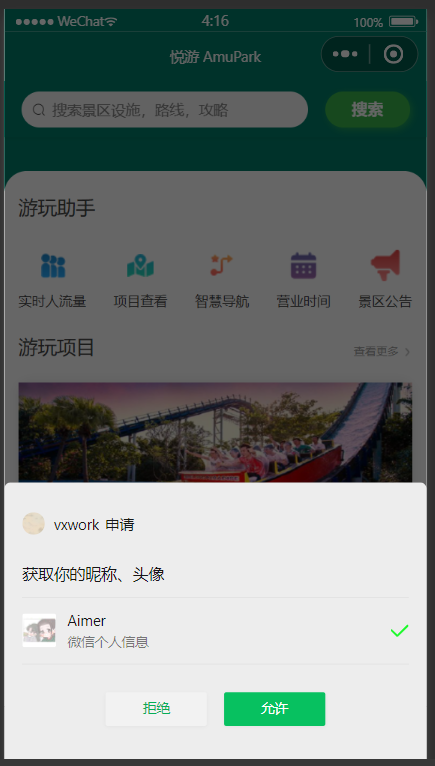
},

fail:function(err){

console.log(err);

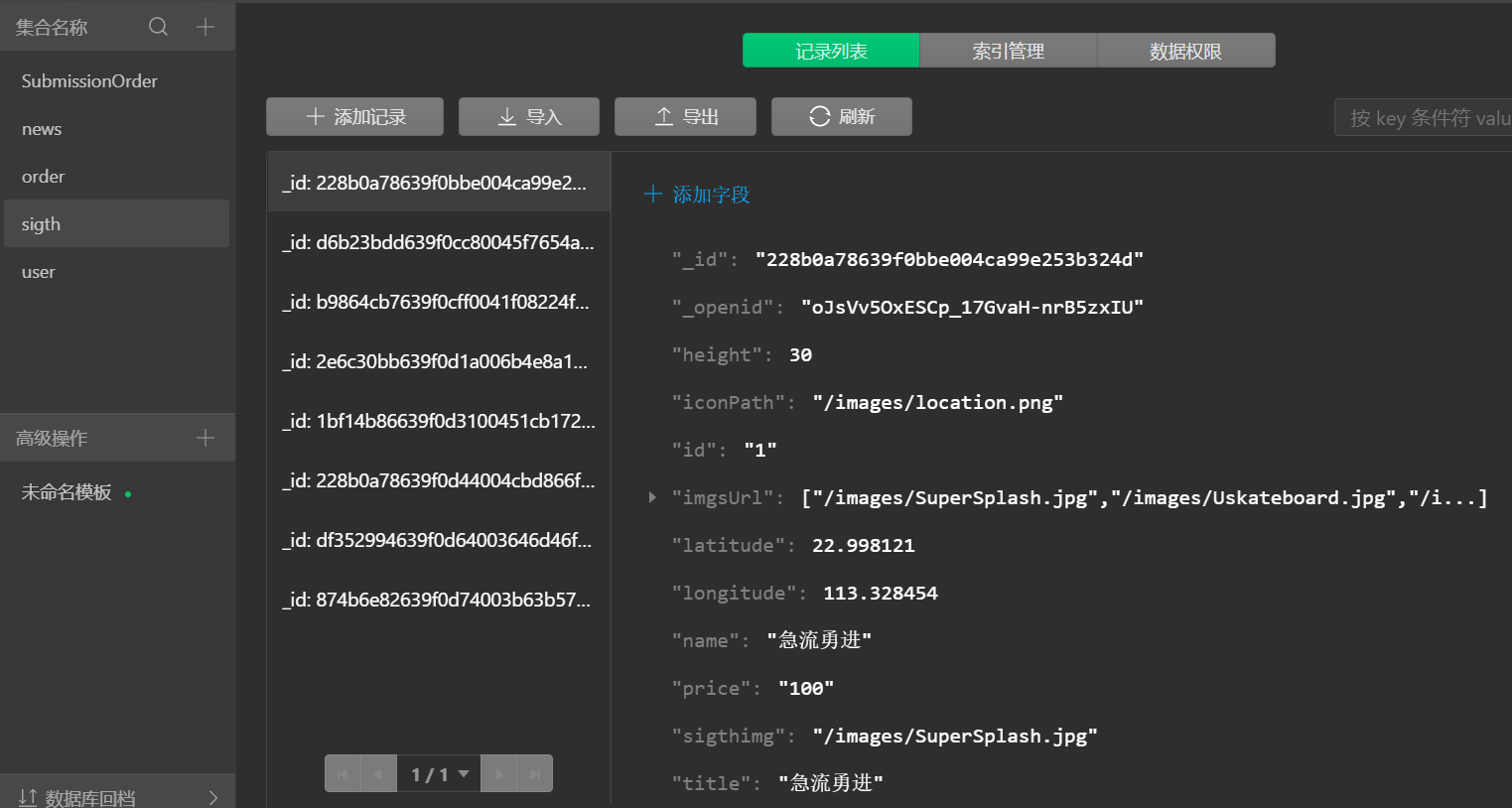
}

})



授权登录页弹窗请求获取用户信息

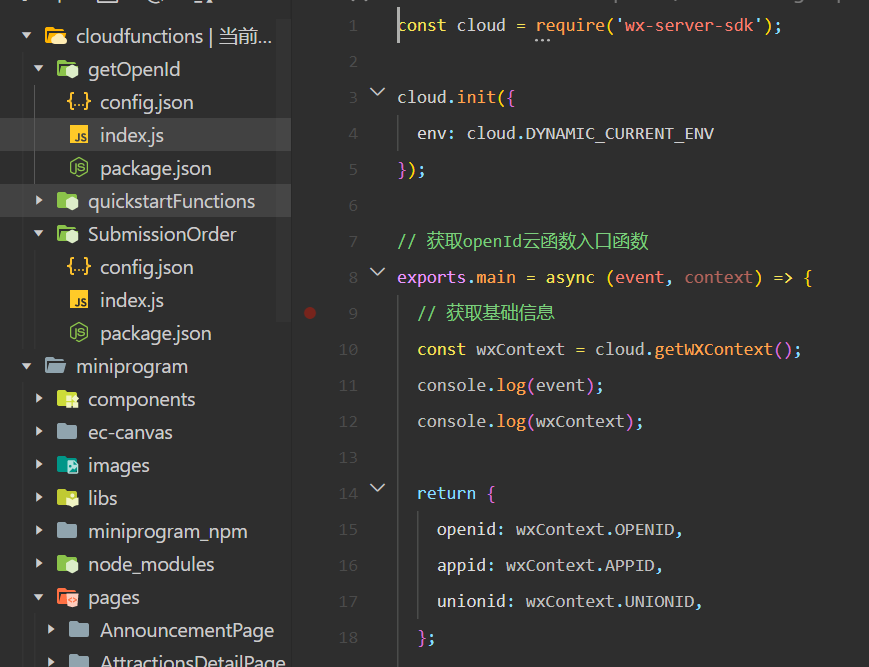
### 5.7小程序云数据库



### 5.8云函数

获取用户openid（该云函数用于根据不同openid显示不同购票订单记录）。

<block wx:for="{{orderlist}}" wx:key="index" wx:if="{{openid}}=={{item.\_openid}}">



### 5.9腾讯地图开放开台

用于显示地图，添加地图标记点及点击事件，实现页面跳转，数据传递。



首页展示景点图片信息，点击对应图片可跳转到同一个页面，显示不同景点信息。



首页：

toSigth(e){

console.log(e)

var sigthId=e.currentTarget.dataset.sigthId;

// console.log(sigthId)

wx.navigateTo({

url: '../SigthDetail/SigthDetail?id='+sigthId,

})

},

SigthDetail页：

console.log(options);

var sigthId=options.id;

console.log(sigthId)

db.collection('sigth').where({

id:sigthId

}).get().then(res=>{

var sigth=res.data[0];

// sigth.data=new Date(sigth.data).toLocaleDateString();

this.setData({

sigth:sigth

})

wx.setNavigationBarTitle({

title: this.data.sigth.title,

})

})

搜索项目：  
 onSigthSearch(){

db.collection('sigth').where({

id:db.RegExp({

regexp:this.data.searchKey,

options:'i'

})

}).get().then(res=>{

console.log(res);

var sigthlist=res.data;

for(let index=0;index<sigthlist.length;index++){

sigthlist[index].date=new Date(sigthlist[index].date).toLocaleDateString();

}

this.setData({

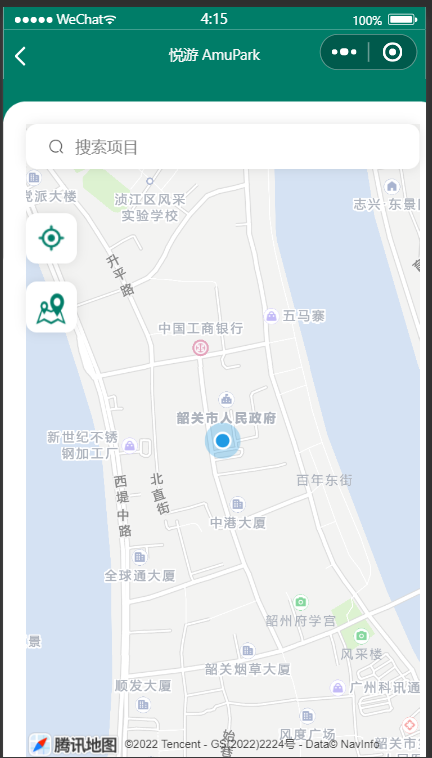
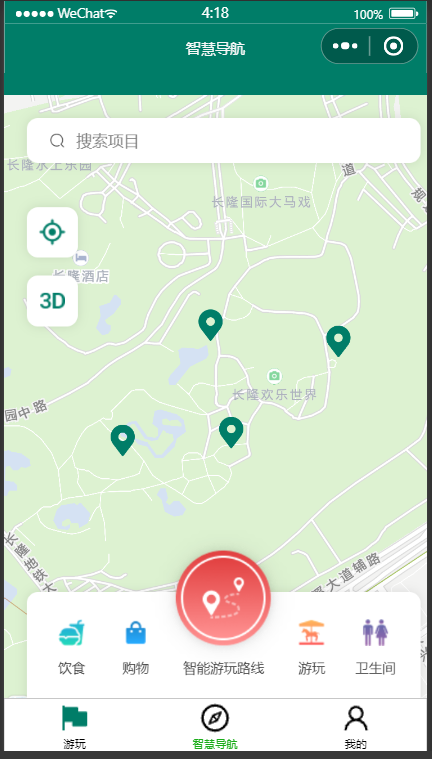
sigthlist:sigthlist

})

})

},

MapPage页显示地图及地图标记点，左上角按钮可定位到当前位置：



// 点击回到原点

moveTolocation: function () {

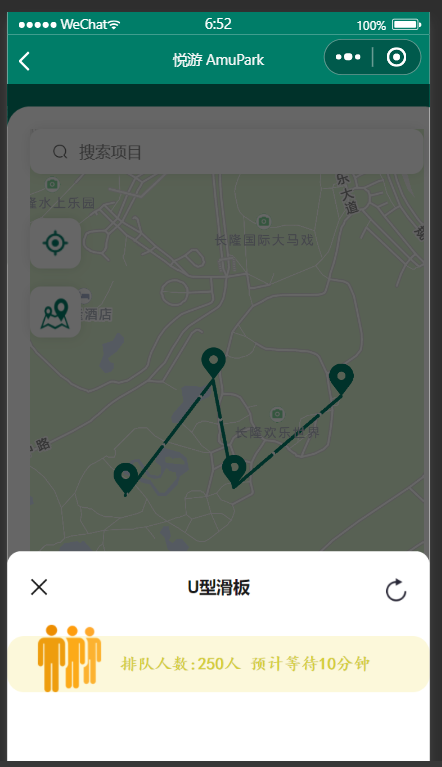
let Id = this.data.mapId

var mapCtx = wx.createMapContext(Id);

mapCtx.moveToLocation();

},

Weui半屏弹窗存放实时数据，点击想查看的景点图标，显示当前景点的排队等候人数及预计等待时间。根据人数不同，图片提示不同。



点击选择图标选择游玩路线：



选择查看路线，弹出半屏弹窗，可左右滑动选择想查看的景点，点击跳到对应页面。返回时地图中心点移动到上一个选择的地图标记点。

<view class="weui-half-screen-dialog\_\_bd" style="padding-top:32px;height:50px;">

<scroll-view class="scroll-view\_H2" scrollX="true" style="width: auto" >

<view class="scroll-view-item\_H2" wx:for="{{markers}}" wx:key="id">

<image class="cover" bindtap="toSigth" data-sigth-id="{{item.id}}" src="{{item.urls}}" data-sigth-latitude="{{item.latitude}}" data-sigth-longitude="{{item.longitude}}"></image>

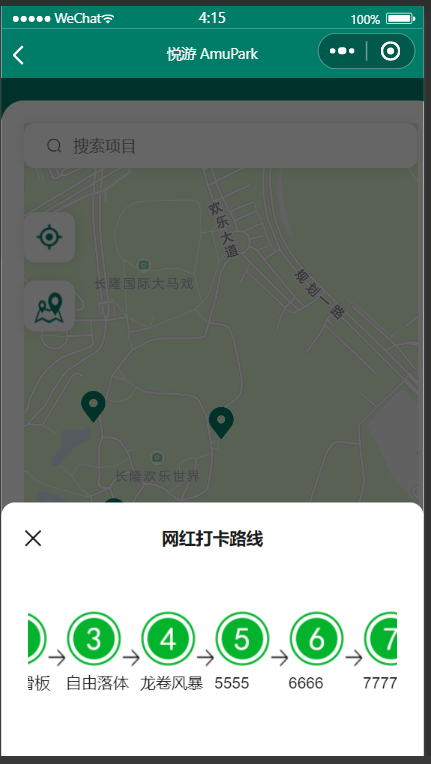
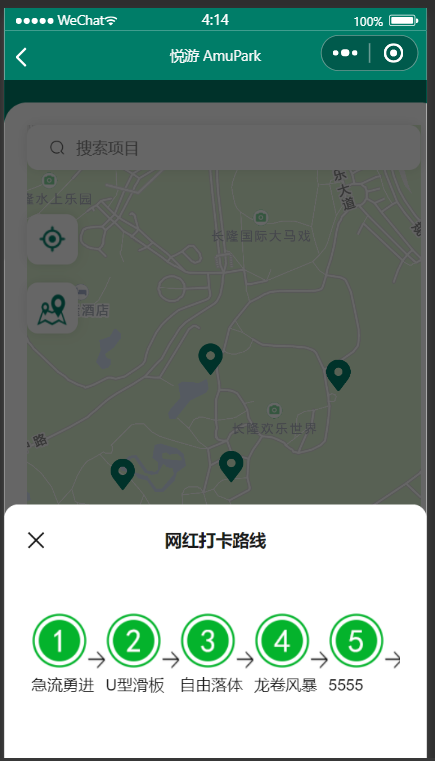
<image class="icon" src="/images/arrow.png"></image>

<view><text>{{item.name}}</text></view>

</view>

</scroll-view>

</view>



购票页面，点击购买跳到对应页面，获取并显示选择的景点名称，填写购买数量，点击确认购买，数据被提交到云数据库中。点击“已购票”，可根据openid查询购票信息。



点击查看详情，可对订单进行操作，删除提交的订单，或者将订单状态改为已完成

