**使用及开发报告**

**AppID**

**wx4ac71de017ebe4f5**

**编写日期 2021年5月**

目录

[第一章 程序概述 1](#_Toc72606457)

[1.1 小程序说明 1](#_Toc72606458)

[1.1.1 灵感来源 1](#_Toc72606459)

[1.1.2 小程序简介 1](#_Toc72606460)

[1.2 需求分析 2](#_Toc72606461)

[1.2.1 问卷调查分析 2](#_Toc72606462)

[1.2.2 用户画像 3](#_Toc72606463)

[1.2.2 应用场景 4](#_Toc72606464)

[1.3 程序特色 4](#_Toc72606465)

[1.4 运营方案 4](#_Toc72606466)

[第二章 功能介绍 5](#_Toc72606467)

[2.1 整体介绍 5](#_Toc72606468)

[2.2 探索板块 6](#_Toc72606469)

[2.3 心路板块 6](#_Toc72606470)

[2.3.1 查看和管理足迹 6](#_Toc72606471)

[2.3.2 发表和编辑足迹 7](#_Toc72606472)

[2.4 用户板块 8](#_Toc72606473)

[2.4.1 用户界面 8](#_Toc72606474)

[2.4.2 用户景点管理界面 8](#_Toc72606475)

[2.5 引导页面 9](#_Toc72606476)

[第三章 界面设计 10](#_Toc72606477)

[3.1动效 10](#_Toc72606478)

[3.2整体UI布局 11](#_Toc72606479)

[3.3配色 12](#_Toc72606480)

[第四章 团队介绍 12](#_Toc72606481)

[4.1 成员及分工 12](#_Toc72606482)

[4.2 时间推进表 12](#_Toc72606483)

[第五章 软件架构和开发环境 13](#_Toc72606484)

[5.1 软件架构 13](#_Toc72606485)

[5.1.1 初始化模块 13](#_Toc72606486)

[5.1.2 景点处理模块 14](#_Toc72606487)

[5.1.3 足迹处理模块 15](#_Toc72606488)

[5.1.4 用户处理模块 16](#_Toc72606489)

[5.2 数据库 16](#_Toc72606490)

[5.3 开发环境 18](#_Toc72606491)

[第六章 可行性分析和软件测试 19](#_Toc72606492)

[6.1 技术可行性 19](#_Toc72606493)

[6.2 经济可行性 20](#_Toc72606494)

[6.3 软件测试 20](#_Toc72606495)

[附录：参考资料 22](#_Toc72606496)

第一章 程序概述

1.1 小程序说明

1.1.1 灵感来源

随着人民生活越来越好，人民越来越富裕。一场说走就走的旅行对我们来说不再是遥不可及，越来越多的人会在节假日去往各个地方的景点游玩。



然而，在旅游之前，对景点的选择也是很大一部分人的纠结所在：想要去游玩的地区有哪些景点？这些景点各自的区别及特色又分别是什么？想要去的景点是否合适自己？

此外，在旅游之后，也有不少人喜欢将旅游中的所见所感记录下来供日后回味。

针对以上情形，本团队希望帮助用户更好更快更清楚地了解到各地区的景点信息，如景点介绍、景点特色、景点内容等，帮助用户快速定位目的地，区分景点是否符合自己的口味，也会为用户提供记录旅游日记的功能。我们还会为用户建立可视化的旅游时间轴，在时间轴上，可以查看自己曾经到达过的景点，感慨过的心情。还有很多有意思的小功能提供给用户，为用户的旅程提供方便快捷走心的服务。

1.1.2 小程序简介

心行FindMyself是一款提供旅游景点介绍和记录旅行日记功能的小程序。本程序致力于为用户提供多样的景点信息，以供用户发现新奇有趣的旅游去向。在用户旅行过程中，本程序还支持用户记录旅行时的所感所得，形成由旅行足迹和旅行日志串联成的时间轴，以便用户回顾自我的旅游过程，在旅行中更好地随心而行，发现旅行之美和自我真谛。

用户进入本程序后，可以打开探索板块，选择省市之后，将会展示一系列含景点的精美图片和简介的卡片，上划后可以看到该景点的详细介绍，包括景点的特色等。用户可以右滑卡片，将其标注并收入“我喜欢的景点”中，或左滑卡片，将其标注为不感兴趣，或者点击跳过，暂时不作取舍。之后，可以在“我”界面中再次查看自己标注过的景点。

用户可以记载旅行中的见闻和心得，写成图文并茂、排版灵活的旅行日志，并写下旅行地点。本程序会将用户写好的日记和相应的旅行地点按照时间由近到远记录成一条时间轴，方便用户日后回顾自己的旅行经历，回味自己的旅行心迹。用户也可以再次修改或删除自己的日志，根据自己的喜好修订好一条属于自己的完美旅行足迹，从中获得新的体会。

为方便用户更好地体验，在本程序中，配备了详尽的使用教程供用户参考。除此之外，还有一些小功能，比如用户可以编辑自己的个性旅游签名，彰显自己的旅游个性；用户也可以管理标记过的和去过的景点，撤销对景点的喜欢或不喜欢标签，重新发现景点不一样的美；另外，在程序界面还附有旅行相关的小贴士，可以让用户收获旅游相关的小知识。

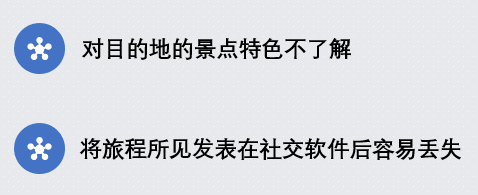
1.2 需求分析

1.2.1 问卷调查分析

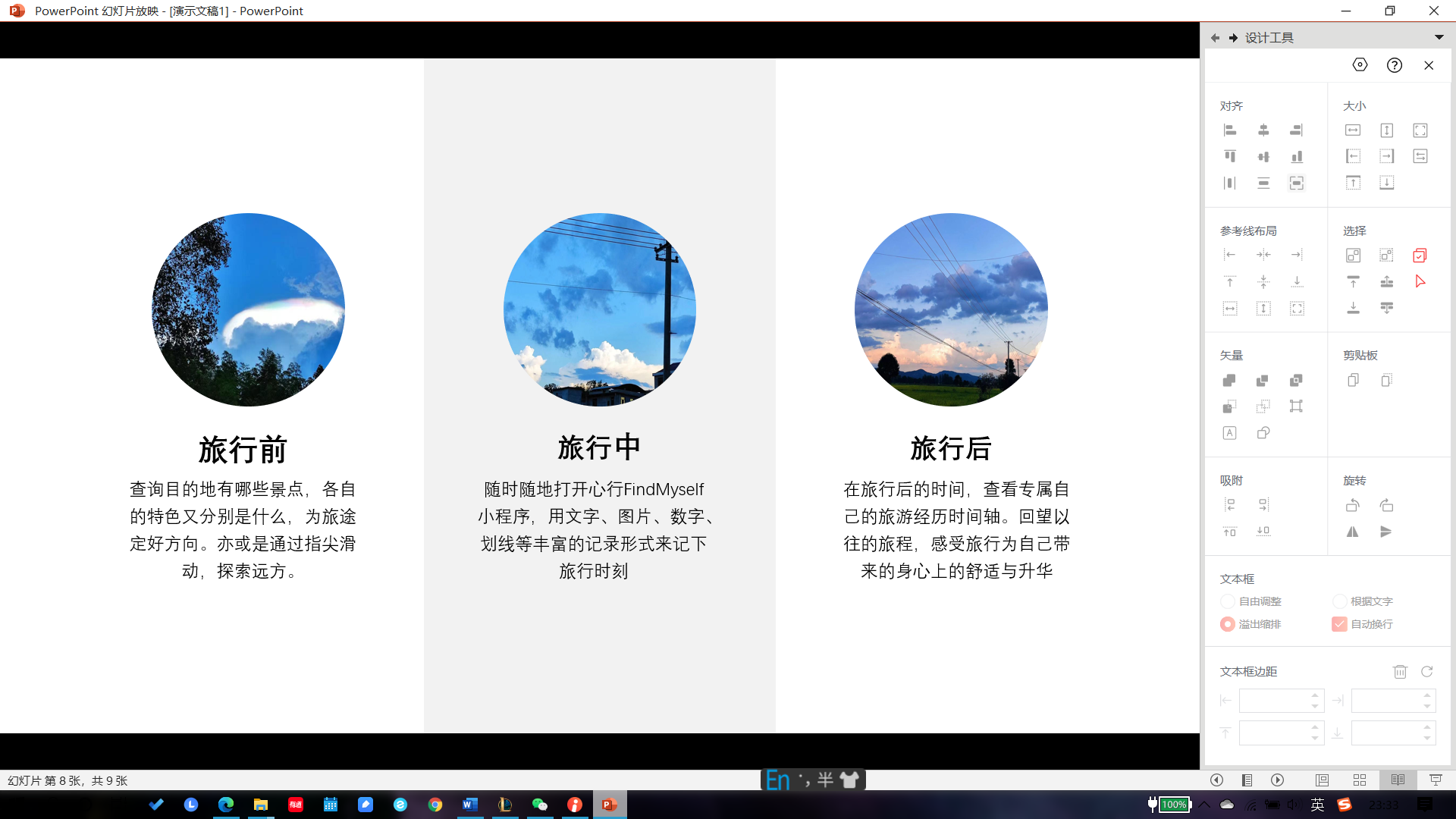
为了更好地了解到广大人民的喜好，我们开展了一次问卷调查，有效填写次数为61人，年龄段在18~45之间，结果如下：

经过调查后发现：

1. 有效填写人数中98.56%的被调查人员喜欢/偶尔想去旅游。显示了旅游这一行为在大众群体中的普遍性，为本小程序在人民群众中的被使用率提供了可靠的参考数据。
2. 有效填写人数中54.1%的被调查人员会在旅程后写下日记来记录该次旅程，86.89%的被调查人员希望拥有自己的旅游经历的时间轴。这两点数据较充分地证明了本小程序的功能被人群所需要。
3. 在调查问卷中，还发现了以下问题



1.2.2 用户画像

1.2.2 应用场景

1.3 程序特色

分锅(大概是几方面：多样化的景点介绍、支持丰富的旅游日志、方便快捷的景点管理)

1.4 运营方案

分锅（大概是第一阶段：暂时只引入广东省景点，未来阶段：引入全国景点乃至世界各地的景点；其他的东西参考一下两篇范文）

第二章 功能介绍

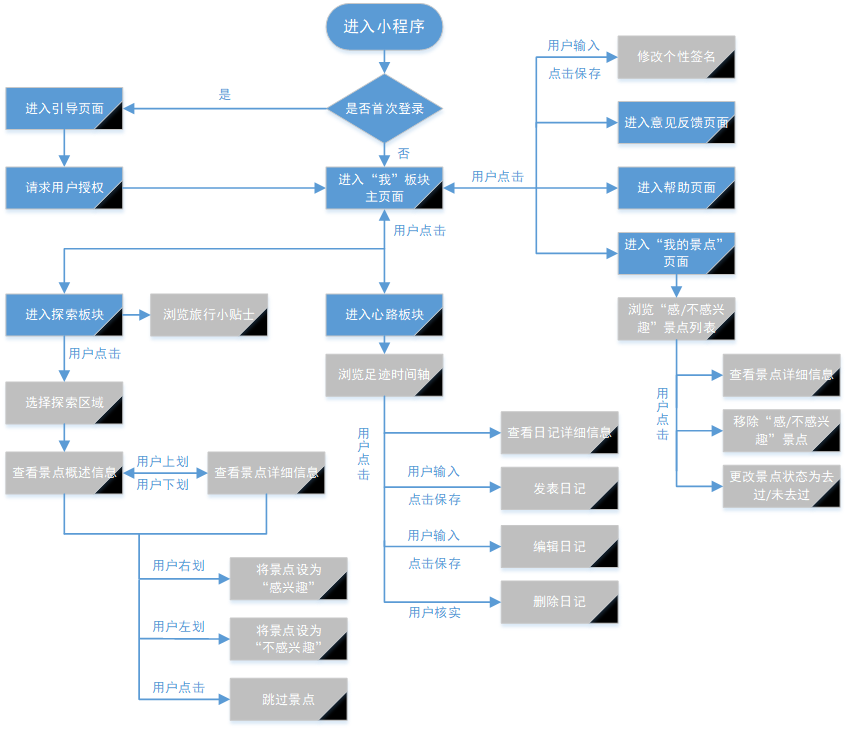
2.1 整体介绍

本程序由探索、心路和“我”三个板块构成。

探索板块包含筛选景点、查看景点简介及详细介绍、旅行小贴士和标注景点的感兴趣与否的功能。

心路板块包含旅行日志的时间轴概览、日志详情查看、发表日志、编辑日志和删除日志的功能。

“我”用户板块包括用户信息展示、个性签名展示和编辑、“我的景点”的查看与重新标注喜好及是否去过、帮助界面、新用户引导教程和反馈界面。

整体框架如下图所示：

2.2 探索板块

在探索板块，可以页面上方有两栏选择栏，分别是省选择栏和市选择栏，用户可以在这里左右滑动查看更多选项。确定自己想要查看的省市区域后，用户可以点击选定省市区域，然后程序会确认更改当前选择的区域并加载出用户选择的区域下的景点卡片。

景点卡片由景点名称、景点图片、景点评分、景点特色和景点地点组成。用户可以上划卡片，将会显示景点的更多信息，新的内容包括景点详细介绍、景点资质和景点细项目分布，同时会收起图片展示。用户可以右滑卡片，代表对这个景点感兴趣；也可以左滑卡片，代表对这个景点不感兴趣。如果难以抉择，可以点击右上角的跳过按钮。这三种操作之后，都会展示下一张卡片。如果当前区域的所有景点都展示完毕了，就会显示如图x所示的提示。

被用户选择了喜欢或不喜欢的景点，都会存储到“我”板块的“我的景点”中可以再次查看和编辑，并且再次选择同一个区域时，喜欢或不喜欢的以及已经去过（在用户和心路板块可以设置已经去过）的景点都不会再次显示，被跳过的景点会再次显示。

在页面下方，会有随机展示的旅行小贴士和其他旅行相关的标语，如图x所示。

所包含的功能：

* 浏览旅行小贴士
* 选择探索区域
* 查看景点详细/概述信息
* 设置景点为感兴趣/不感兴趣
* 跳过景点

（截图建议：刚进到这个界面图2-2-1，滑动一下城市选择栏(不点击选择)图2-2-2，界面(点击选择后)2-2-3，上划详细信息2-2-4，左划中(卡片动画未完全划走，下同)2-2-5，右滑中2-2-6，跳过(如果按钮加了动效会更有展示价值)2-2-7，小贴士(p图圈一下)2-2-8，请把所有上文的如图x的x换成截图后你所使用的图片编号）

2.3 心路板块

2.3.1 查看和管理足迹

在心路模块，记载用户的旅行日记和旅行过的景点。时间轴展示的内容单元是足迹，足迹是无日志的已去过一个景点或景点的一篇旅行日志。足迹中景点和日志的样式不相同，便于区分。所有的足迹按照时间的由近到远展示。为保护隐私，在本程序内，所有足迹都只有用户自己可见，其他用户无法查看。

初始时，足迹会展示景点的名称、足迹的日期。如果足迹是一篇日志，右方会有展开按钮。当用户点击右方的展开按钮后，就会详细展开这篇日志的内容，按钮会变为收起按钮。点击收起按钮会恢复为原来的状态。足迹的右下方还有编辑和删除的图标按钮。用户点击删除按钮后，为防止误触，程序会弹出一个确认框，用户点击确认后，将会删除这个足迹。用户点击取消后，将不会触发删除。

当用户没有记载一篇日记也没有去过一个景点时，页面会显示如图x所示。

所包含的功能：

* 查看足迹时间轴
* 查看日记详细信息
* 删除足迹

（截图建议：(本来有2篇日记，一篇其他景点，一篇打卡景点，和一个已去过的无日记足迹(在我的景点点击去过可以增加这样的足迹)，随便一篇日记写点正经的内容，包含图片，H1~6其中一个+居中，包含加粗，下划线等)，一开始到这个页面截图2-3-1-1，点击展开写了正经内容的日记截图2-3-1-2，点击删除(删掉不是正经日记的一篇日记)弹窗截图2-3-1-3，确认删除后截图2-3-1-4）

2.3.2 发表和编辑足迹

当用户点击页面右上角的发帖按钮时，将会进入发帖页面。用户可以选择一个在探索页面已有的景点，为该景点写一篇景点打卡旅行日志，此时，景点的名字会默认设置为打卡点的名字，用户也可以自行更改。用户也可以选择自定义，为探索页面外的其他景点撰写自己的旅行日志，并附上景点的名称。旅行日记的日期将会设置为发表日记的当天。

对于发表日志的正文，用户可以选择丰富的排版格式，例如6个等级的标题、下划线、加粗、斜体、删除线、居中、设置或取消缩进、分割线、快速输入今天的日期，还可以在任意地方插入图片。如果用户不输入日记标题，程序会根据景点名字为用户智能补充一个名为该景点的标题。完成日记记录之后，用户可以点击提交按钮发布自己的旅行日志，发布后马上可以在时间轴顶部看到自己新的足迹。用户也可以点击右上角的返回按钮或手机的返回键，丢弃当前的内容，取消日记发表。

当用户点击编辑按钮后，会跳转到发帖页面，如果本来是无日志的景点，那么将为该景点添加一篇日志；如果是日志，那么将在编辑框内显示该日志本来的所有内容，用户可以选择对该日志在原有基础上进行修改。点击确认后保存更改，点击右上角的返回按钮或手机返回键后，将丢弃当前修改。

所包含的功能：

* 发表日记
* 编辑日记

（截图建议：进入发帖页面截图2-3-2-1，选择打卡景点点开2-3-2-2，选一个非首个市和非首个市，截图2-3-2-3，选择其他，截图2-3-2-4，在2-3-2-3的基础上点击确认，输入日记标题，截图2-3-2-5，选择自定义，截图2-3-2-6，用H1~H6各写一行字，截图2-3-2-7，使用下划线、加粗、斜体、删除线(含单独和复合使用)，日期，居中，分割线，截图2-3-2-8，使用增加首行缩进，两张图片，图文混拍，截图2-3-2-9，缩放图片，截图2-3-2-10，发布后展开刚发布的日记，截图2-3-2-11，编辑之前正经的日记弹出编辑发帖的页面后，截图2-3-2-12，随便改点什么，可包括标题，打卡景点，保存，展开编辑过的日记(要看得出编辑过)，截图2-3-2-13）

2.4 用户板块

2.4.1 用户界面

用户首次进入“我”用户板块时，会显示一个登录按钮，用户点击后会弹出授权窗口，当用户确认授权时，程序会读取用户的头像和微信昵称，并为用户自动建立账号，用于存储探索板块和心路板块用户的相关记录，将会进入用户板块的主要界面。如果用户不进行授权，“我”页面保持该状态。

在主要界面中，上方会显示用户的头像，昵称还有旅行个性签名。初始状态下用户没有旅行个性签名，可以点击编辑按钮进行设置，签名长度不能超过20字，设置完毕后点击保存按钮即可保存旅行个性签名。在这之后将会显示用户自己设置的旅行个性签名。

在下方会有三个栏目，分别是“我的景点”用户景点管理界面，意见反馈和帮助页面。

在意见反馈栏目，用户可以向开发者反馈自己在使用过程中可能会遇到的问题，提出自己对本程序宝贵的意见，也可以向微信投诉和举报。

在帮助栏目，用户可以查看本程序的详细使用流程介绍。

所包含的功能：

* 修改旅行个性签名
* 进入帮助页面
* 进入意见和反馈页面
* 进入“我的景点”页面

（截图建议：(本来没有个性签名)进入此页面，截图2-4-1-1，点击编辑按钮，截图2-4-1-2，随便改点什么，截图2-4-1-3，保存，截图2-4-1-4，点击意见反馈页，截图2-4-1-5，点击帮助页，截图2-4-1-6，往下拉到最下面，截图2-4-1-7）

2.4.2 用户景点管理界面

在“我的景点”用户景点管理界面，用户可以查看自己已经标注感兴趣的或不感兴趣的景点列表。当没有标注感兴趣的或没有不感兴趣的景点时，程序将会显示如图x所示。

当用户在上方选定“感兴趣的”或“不感兴趣”时，页面将会列表罗列出所有用户已经标注的感兴趣的或不感兴趣的景点横条，显示景点的名字和用户是否已经去过该景点。如果用户发表过该景点的旅行日志，将会认为是已经去过该景点。也可以编辑景点更改是否去过的状态。横条左边的图标表示是否去过的状态，如果图标是问号，代表未去过，如果是勾，代表去过。

用户可以点击横条，查看该景点的详细信息，这包括景点的图片、地点、评分、介绍、特色、资质和项目分布。用户可以上下滑动详细信息。点击页面上方阴影部分后将回到本来的状态。

用户可以右拉一个景点横条，展开管理景点的功能按钮。左边的按钮用于管理一个景点是否去过，图标状态与该景点原本状态相反，代表点击后将修改为什么状态。该按钮的图标为勾时，代表该景点未去过，点击后设置为已去过；该按钮图标为问号时，代表该景点已去过，点击后设置为未去过。点击按钮后或左滑后，将收起管理景点的功能按钮。

对于未去过的景点，点击按钮修改为去过之后，状态将会显示更改，并且在心路板块可以看到一条新的足迹。对于已经去过的景点，用户可以重新将其设置为未去过，设置之后，状态同样会更改，并且在心路板块可以看到新建的足迹被撤销。

右边的减号图标的按钮是移除按钮，点击后，用户将重新编辑对景点的感兴趣与否状态，可以取消设置某个景点的感兴趣/不感兴趣标记，此时，该景点会移出这个列表，并且在探索板块可以重新看到这个景点。

所包含的功能：

* 浏览感兴趣/不感兴趣的景点列表
* 查看景点详细信息
* 移除感兴趣/不感兴趣的景点
* 更改景点状态为去过/未去过

（截图建议：进入此页面(本来有若干个感兴趣，有些去过有些没去过的景点，无不感兴趣的景点)，截图2-4-2-1，展开一个景点，截图2-4-2-2，下拉到详细信息底部，展开2-4-2-3，右滑一个景点2-4-2-4，点击左边按钮，弹窗更改成功时截图2-4-2-5，点击右边按钮，点击后截图2-4-2-6，滑到不感兴趣，截图2-4-2-7）

2.5 引导页面

用户首次进入本程序时，会显示引导页面，该页面有三幅滚动播放的图片，简要地介绍了程序的功能。用户也可以自由滑动查看这三张图片。在页面下方，有一个点击按钮。用户点击之后将跳出引导界面，进入“我”用户板块。之后，用户可以点击登录按钮，授权登录本程序，登录后将进入用户页面。之后每次进入程序将会跳过引导界面，直接进入用户页面。

（截图建议：三个滑动栏都截图2-5-1 ~ 2-5-3，点击按钮动效时截图2-5-4，跳到我界面后截图2-5-5，点击登录后弹窗截图2-5-6）

第三章 界面设计

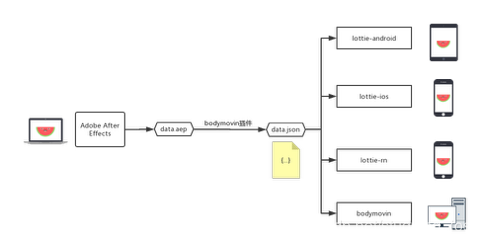
3.1动效

为了提升本小程序为使用者带来的愉快体验，我们引入了Lottie动效的使用。

🞈什么是Lottie？

Lottie是Airbnb开源的一个动画渲染库，支持多平台，包括iOS、Android、React Native以及Flutter。

🞈整体工作流程为：



1）动效设计师使用After Effects制作动画，然后使用Bodymovin导出JSON文件，可以将JSON文件放到Bodymovin网站上运行看效果，也可以放在lottiefiles网站上运行看效果，而且lottiefiles有很多免费动画JSON资源可以下载看。

2）各个端使用对应的LottieSDK加载JSON文件，实现动画效果。本小程序是通过微信官方Lottie接口来调用lottie的json文件，从而实现观赏性极强的动画效果

**此外，由于使用了Lottie技术，动效的显示便不依赖于Gif图片，从而在一定程度上提升了小程序的性能，减少所占空间。**

3.2整体UI布局

* 页面结构

1.“我页面”

2.“探索页面”



1. （2）

如图（1）所示，在本程序的“探索”模块中主要分为三个模块：

1. 区域选择板块：在这个模块中，用户可选择目的地所在的省、市，从而达到查询目的地各景点特色的目的。在此模块的UI设计考量上，我们采用了**可视动效化**的滑动多列选择轴。蓝色底线将会滑向被选择的省，而绿色底线将会滑向被选择的市。
2. 景点介绍模块：在这个板块中，各景点将会以卡片的形式呈现出来，在卡片正面有景点名称、景点图片、景点评分、景点特色、景点地点、旅游短句的呈现，而通过手指向上滑动，则可进入卡片背面，在卡片背面有更为详细的景点介绍。更为重要的是，用户在浏览完景点卡片后，可通过左滑不感兴趣、右滑感兴趣来选择景点对自己是否有足够的吸引力，滑动后，卡片将会出现左滑/右滑的动效。
3. 动效呈现模块：除了卡片的动效外，本模块也会呈现Lottie动效来为用户带来更愉快的使用体验。

3.3配色

第四章 团队介绍

4.1 成员及分工

覃梓鑫

任务安排、统筹管理

大部分逻辑部分编写

界面框架设计

文档撰写

杨佳欣

需求分析

其余逻辑部分编写

UI设计

文档撰写

管晓涛

景点数据收集和处理

其余逻辑部分编写

UI设计

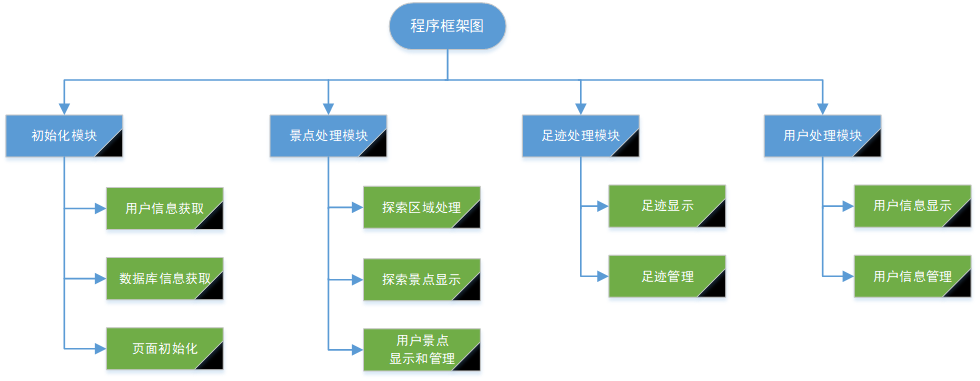
文档撰写

4.2 时间推进表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 3月上旬 | 3月中旬 | 3月下旬 | 4月上旬 | 4月中旬 | 4月下旬 | 5月上旬 | 5月中旬 | 5月下旬 |
| 需求分析 | 3月5日~3月20日 初步分析和问卷调查  3月22日~4月3日 最后确认需求 | | | | | | | | |
| 可行性分析 | 3月18日~4月7日 相关资料收集  4月10日~4月13日 确定初步方案  4月20日 确定第二版方案 | | | | | | | | |
| 小程序开发 | 4月27日~5月4日 核心内容设计  5月6日~5月15日 UI设计、剩余部分设计 | | | | | | | | |
| 小程序改进 | 5月6日~5月15日 优化程序功能 | | | | | | | | |
| 单元测试 | 5月3日~5月18日 修复漏洞、代码覆盖测试  5月19日~5月20日 性能测试、内测 | | | | | | | | |
| 制作视频 | 5月18日~5月23日 视频制作和修改 | | | | | | | | |
| 制作介绍文档 | 5月13日~5月23日 文档制作和修改 | | | | | | | | |

第五章 软件架构和开发环境

5.1 软件架构

本程序主要由初始化模块、景点处理模块、足迹处理模块、用户处理模块构成。整体架构如图：

5.1.1 初始化模块

初始化分为用户信息获取、数据库信息获取、页面初始化三部分。

用户信息获取的流程如下：

1. 调用云函数，获取用户的openid
2. 根据openid查询云开发数据库user集合中是否有该用户；如果没有，加载引导页面和授权登录按钮；如果有，读取用户信息，进入“我”板块主页面
3. 如果第二步数据库有用户，读取该用户的所有足迹，进行相关页面初始化
4. 如果第二步数据库无用户，等待用户主动点击授权获取信息，获取成功后创建数据库记录。

数据库信息获取流程如下：

1. 初始化云开发功能
2. 读取云开发数据库global集合，获取当前省份数、城市数、景点数
3. 依据上一步得到的数目，按照每次读取20个数据的方式同时批量读取景点、城市、景点集合，优化读取速度。
4. 异步检测到全部数据读取完毕后，对接收到的数据重排序，进行相关页面初始化，

注：所有信息一次性加载完毕，优点在于不用每次点击后再读取数据库，避免点击后因网络延迟带来的卡顿而给用户不良的体验。

页面初始化：

1. 页面初始化完毕之前，禁用用户对页面的点击操作
2. 如果页面加载完毕时所需数据尚未成功接收，向app.js传递一个函数，待接收成功后，由app.js执行该函数，完成页面所需数据载入和显示。
3. 如果页面加载完毕前数据已经成功接收，直接将相关数据载入和显示。
4. 页面初始化完毕或因网络缘故加载失败，解除对用户对页面的点击操作的禁止；如果是加载失败，提示用户重新进入程序

5.1.2 景点处理模块

景点处理模块分为探索区域处理、探索景点显示、用户景点显示和管理三部分：

探索区域处理流程如下：

1. 在初始化模块中得到相关信息，包括用户感兴趣的景点、不感兴趣的景点和已经去过的景点，以及省份、城市、景点的数据
2. 随机展示一条旅行小贴士
3. 等待用户操作，获取用户需要的探索区域
4. 根据用户选择的探索区域，并排除掉用户感兴趣的景点、不感兴趣的景点和已经去过的景点，筛选出用户未曾标记的景点供用户探索，执行探索景点显示处理流程
5. 如果用户重新选择探索区域，回到第二步

探索景点显示处理流程如下：

1. 在探索区域处理的第三步，得到所有符合条件的景点
2. 将景点打乱顺序
3. 将当前第一个景点显示在页面上，加载图片
4. 等待用户操作，如果用户选择左划、左划或跳过，显示下一个景点
5. 如果所有景点均显示完毕，向用户显示当前区域探索完毕

用户景点显示和管理处理流程如下：

1. 在初始化模块中得到相关信息，包括用户感兴趣的景点、不感兴趣的景点和已经去过的景点，以及数据库中景点的数据
2. 用户打开“我的景点”页面，选择“感兴趣”（进入页面时自动选择该项）或“不感兴趣”的景点，将用户选择的景点列表展示在页面上
3. 如果用户点击展开或折叠，更改显示状态，加载图片
4. 如果用户点击取消设置一个景点“感兴趣”或“不感兴趣”，将其在列表中删除，并将改变写入数据库和通知探索区域流程。
5. 如果用户点击设置一个景点为“去过”，更改页面显示状态，写入一个新的足迹，将改变写入数据库和通知足迹处理模块。
6. 如果用户点击设置一个景点为“未去过”，更改页面显示状态，如果查找到由“去过”状态写入的足迹，删除这些足迹，将改变写入数据库和通知足迹处理模块。

注：在程序流程中，所有涉及写入数据库操作的按钮，在页面都会禁止用户频繁点击，点击按钮后改变状态，禁止用户再次点击，直到所有对数据库的异步操作处理结束或因网络缘故处理失败，恢复按钮可点击。如果是网络缘故，提示用户再次操作。

5.1.3 足迹处理模块

足迹处理模块分为足迹展示和足迹管理两部分。

足迹展示处理流程如下：

1. 在初始化模块中得到用户的所有足迹信息。
2. 将足迹以时间顺序由近到远排序，并展示在页面上
3. 如果用户展开一个带日记的足迹，调用富文本编辑器解析日记内容的WXML代码，加载图片，将结果展示在页面上；如果用户折叠足迹，隐藏结果。

足迹管理处理流程如下：

1. 如果用户删除一个足迹，首先询问用户是否确认，如果用户取消，不做改变；如果用户确认，则在用户的足迹数组中删除掉该元素，以确保用户无法再次加载该日志，实现了删除操作，将结果写入数据库，并通知用户景点显示和管理处理流程。
2. 如果用户新建一个日记足迹：
   1. 进入新页面，根据当前日记总数申请一个唯一的日记\_id，将日记总数临时自增，写入数据库。等待用户完成所有操作。
   2. 如果用户按下返回，取消记录操作，日记总数临时自减，写入数据库。
   3. 如果用户每次上传图片，用于计序数的变量自增，调用API，上传图片，将用户上传的图片写入云开发云存储中。
   4. 如果用户删除图片，序数变量不变，以确保反复增删时不会产生序号混乱。
   5. 如果用户按下提交，将用户记录的日记足迹写入数据库，并通知用户景点显示和管理处理流程。
3. 如果用户编辑一个日记足迹：
   1. 进入第二步的页面，将原日记足迹内容加载在编辑框中，日记总数不会临时自增
   2. 使用正则表达式处理WXML，统计原有多少图片，以确定图片序数
   3. 其他处理步骤与第二步相同，但用户按下返回后日记总数不会临时自减

5.1.4 用户处理模块

用户处理模块分为用户信息展示和用户信息管理两部分。

用户信息展示流程如下：

1. 在初始化模块中得到用户信息
2. 将用户信息加载在页面上，如果用户没有旅行个性签名，显示一段提示文本
3. 如果在其他处理流程中，用户信息发生了改变，重新执行第二步

用户信息处理流程如下：

1. 如果用户点击编辑旅行个性签名，将旅行个性签名展示框换成编辑框，编辑框初始值为原本的个性签名；用户点击保存后将改变显示并写入数据库
2. 如果用户点击帮助、意见反馈页面，进入相应页面

5.2 数据库

本程序使用了微信小程序的云开发，数据库环境ID为lr580c-6gotth6z00871312，有效集合有province、city、attration、user、diary、global。

province集合存储各省级地区信息，city集合存储各市级地区信息，attration集合存储各景点信息。省、市、景点形成的从属包含关系，即一个省包含一个到多个多个市、一个市包含一个到景点；且一个景点只属于一个市，一个市只属于一个省，所以它们还是树状的关系。

user集合存储用户信息，diary集合存储用户足迹信息，其中一个用户对应零个到多个足迹，且一个足迹只属于一个用户，这保证了足迹只可以被用户自己看见。

global集合存储全局信息，只有一个\_id为default的记录，其字段和说明列表如下：

|  |  |
| --- | --- |
| 字段名 | 说明 |
| \_id | default |
| num\_province | 当前省份数目 |
| num\_city | 当前城市数目 |
| num\_attration | 当前景点数目 |
| num\_diary | 当前日记数目 |

province集合的有用字段和说明列表如下：

|  |  |
| --- | --- |
| 字段名 | 说明 |
| \_id | 从0开始计数 |
| name | 省份的名称 |
| cities | province记录中属于该省的\_id数组 |

city集合的有用字段和说明列表如下：

|  |  |
| --- | --- |
| 字段名 | 说明 |
| \_id | 从0开始计数 |
| name | 城市的名称 |
| attration | attration记录中属于该市的\_id数组 |
| belong | 该市所属的省份\_id |

attration集合的有用字段和说明列表如下：

|  |  |
| --- | --- |
| 字段名 | 说明 |
| \_id | 从0开始计数 |
| belong | 该市所属的城市\_id |
| name | 景点名称 |
| intro | 景点介绍 |
| pic | 景点图片URL |
| position | 景点位置 |
| mark | 景点评分 |
| detail | 景点旅游项目分布字符串数组 |
| prize | 景点资质字符串数组 |
| special | 景点特色字符串数组 |

user集合的有用字段和说明列表如下：

|  |  |
| --- | --- |
| 字段名 | 说明 |
| \_id | 与\_openid一致 |
| avatarUrl | 用户头像URL链接 |
| nickName | 用户微信昵称 |
| motto | 用户旅行个性签名 |
| like | 用户感兴趣的景点\_id数组 |
| dislike | 用户不感兴趣的景点\_id数组 |
| gone | 用户去过的景点\_id数组 |
| diary | 用户的足迹\_id数组 |

diary集合的有用字段和说明列表如下：

|  |  |
| --- | --- |
| 字段名 | 说明 |
| \_id | 从0开始计数 |
| user | 该足迹所属的用户的\_id |
| content | 日志内容(WXML格式) |
| time | 发表时间 |
| att\_id | 景点\_id |
| att\_name | 景点名称 |

5.3 开发环境

|  |  |
| --- | --- |
| 开发环境 | |
| 开发工具 | Visual studio Code |
| 微信开发者工具 |
| 开发语言 | JavaScript、json等 |
| 图片处理工具 | Photoshop、AfterEfect |

第六章 可行性分析和软件测试

6.1 技术可行性

本程序的技术实现是可行的，分析如下：

1. 在**功能设计**上，所有功能在技术上都是可以实现的，而且充分考虑了各种边界条件，经过了反复严谨的代码测试，具有良好的健壮性。例如，所有功能都支持用户的撤销操作，例如不授权登录、取消编辑中的写日记或编辑日记、取消误点的删除。对于用户输入内容为空时，也具有相应的提示。  
   本程序的代码效率较高，没有冗余的代码和重复多余的操作，并且均在良好的时间复杂度和空间复杂度内实现了代码逻辑，没有出现指数级复杂度的代码逻辑。  
   本程序在保证功能正常运转和高效的同时，还尽可能提供了完善的功能设计。例如在设计探索区域时，用户可以选择景点为喜欢/不喜欢，也可以在“我的景点”页面撤销；用户可以发布旅行日记，也可以编辑和删除旅行日记，而且旅行日记支持丰富的排版，实现机制是使用editor，直接将用户编辑的内容存储为HTML格式，读取日记内容时调用API对HTML内容进行动态渲染，因此可以支持网页能够渲染出的多种多样的排版和样式。
2. 在**数据库操作**上，使用了微信开发者工具自带的强大云开发功能，能够灵活和高效地处理数据，包括载入景点数据、用户信息等。  
   使用云数据库，是因为考虑到用户可能使用不同的设备使用本程序，如果存储在本地，一来更换设备时个人数据如旅行日记会丢失，二来用户定期清理缓存时也有可能会把小程序的数据清理掉。所以，存储用户的信息必须使用云数据库。  
   由于读写云数据库有网络延迟，所以在所有需要读写数据的地方，都使用异步设置了加载提示、防止频繁点击和处理网络故障的相关代码，确保了程序的健壮性，在这里避免了程序崩溃的可能性。  
   为了提高性能，所以每次读取数据库都尽可能读取最大的记录条数，大大减少了加载耗时。由于探索板块的景点数据和一些用户基本信息是保持不变的，并且总数据量不大，所以在程序加载之初，就一次性全部读取，避免每次要更换探索景点区域再重新读取数据库，产生网络延迟，影响用户体验。所以首次加载之后，加载每个页面都不会有网络卡顿现象。
3. 在**内容设计**上，我们综合参考了多方权威可行的资料，然后经过精心的人工筛选、整合、处理，挑选出了极具价值景点信息数据库。通过这样的方法收集景点信息数据，与使用Python爬虫批量获取相比，我们可以避免内容重复度高、良莠不齐、单调、类目不全的问题。因此，本程序可以提供更优质、更有含金量、更丰富、更全面的景点内容，让用户可以在探索板块发现更多心仪的旅游景点。
4. 在**页面设计**上，通过使用Canvas和调用微信API接口等，可以很好地实现美观、动态的页面UI设计。使用相关的代码逻辑判断，综合使用wxss,wxml和js，可以很好地协调实现动画，例如景点卡片的左滑、右滑效果。

6.2 经济可行性

本程序在经济上是可行的，分析如下：

1. 在操作上，本程序**方便快捷**，只需要通过简单和少量的点击既可以快速满足自己的使用需求，如寻找新的心仪景点、管理自己标记的景点、编写和管理旅行日记。用户可以快速理解和操作，节约了用户的使用时间成本，也迎合主要用户群体的需求。
2. 录入了丰富多样的景点数据，并且支持丰富格式的旅行日记编写，能够让用户感受到本程序的魅力，**给用户带来使用上的愉悦，可以有效增加用户粘性和用户活跃时间**。
3. 有丰富的细节，使用了动效UI设计，增加了旅行小贴士和可编辑的个性签名，这些都能够让用户拥有更好的**使用体验**。

6.3 软件测试

我们对本程序进行了较为充分的测试，包括代码覆盖测试、多环境测试和压力测试。

**代码覆盖测试：**

（截图建议：按照操作来(不需要截图的某些操作可以不操作qwq)，看到结果如图x时截个图，然后把如图x的x换成对应的比方如图6-3-1,6-3-2，压力测试同）

测试流程和结果如下：

1. 使用一个从未使用过本程序的微信号，扫描体验码，进入小程序。
2. 进入引导页面，如图x，进行多次反复左滑、右滑，退出小程序。  
   目的：测试动画效果是否流程且验证未完成引导页面退出后重进仍进入引导页面
3. 点击按钮，进入登录页面。
4. 点击登录按钮，取消授权。结果弹出提示框如图x。  
   目的：验证程序允许用户不进行登录。
5. 点击帮助页面，下拉到底，然后返回未登录页面。
6. 点击探索页面，选择一个区域，然后进行多次左滑、右滑、上滑和下滑。  
   目的：验证程序允许用户不进行登录也能使用探索功能。
7. 点击探索页面，发现弹出提示窗口如图x。点击发表日记按钮，发现弹出提示窗口如图x。  
   目的：验证程序允许用户不进行登录不能使用心路功能。
8. 点击登录按钮，进行授权。
9. 点击意见反馈页，点击产品建议，随意填写测试内容然后提交。
10. 点击“我的景点”页，如图x所示。选择“不感兴趣”，结果同图x。
11. 返回，点击探索板块，直接左滑一次，然后上滑一次、左滑一次，然后右滑一次、上滑一次、右滑一次，然后一直点击跳过，直到景点为空，结果如图x所示。  
    目的：验证“探索板块”查看景点基本功能正常。
12. 点击“我”板块，点击“我的景点”，查看“感兴趣的”和“不感兴趣”，发现符合之前操作的结果。结果如图x和图x所示。
13. 点击一个“我的景点”中的景点，查看详细信息，下拉到底，然后返回。
14. 右滑一个景点，点击一次左边按钮。查看状态。
15. 选择打开“不感兴趣”，右滑这两个景点，点击右边按钮。
16. 返回，点击“心路”板块，发现有一个对应的足迹。结果如图x所示。
17. 点击“我”板块，点击第14步操作的景点，右滑，再次点击左边按钮，查看状态。
18. 返回，点击“心路”板块，发现足迹已经被移除。结果如图x所示。  
    目的：验证“我的景点”基本功能正常。
19. 点击“探索”板块，左滑所有景点，逐一检验，发现只有仍在“感兴趣的”景点没有展示。
20. 选择一个区域A，景点卡片进行了更新。选择回原本的区域，景点卡片为空。结果如图x所示。再次选择区域A，景点卡片更新，且初始卡片与上一次不一样。结果如图x所示。  
    目的：验证“探索板块”选择区域功能正常。
21. 点击“我”板块，再回到“探索”板块，发现小贴士发生了改变。结果如图x所示。  
    目的：验证“探索板块”小贴士功能正常。
22. 回到“我”板块，编辑个性签名并保存。结果如图x所示。  
    目的：验证“用户板块”旅行个性签名功能正常。
23. 点击“心路”板块，点击提交按钮，正文输入任意内容，然后点击返回键。
24. 再次点击提交按钮，点击提交，提示内容结果如图x所示。
25. 任意输入内容，使用一次所有格式，上传一张图片，并对图片缩放，再次点击提交按钮。展开日记，如图x所示。
26. 编辑25步的日记，输入任意内容，按下返回，展开日记，发现同图x所示。
27. 再次编辑25步的日记，输入任意内容，修改打卡景点，修改标题，点击提交按钮，展开日记，结果如图x所示。
28. 删除25步的日记，点击取消。
29. 再次删除25的日记，点击确认。结果如图x所示。  
    目的：验证“心路板块”功能正常。
30. 退出小程序，再次进入小程序。
31. 测试完毕，退出小程序。  
    目的：验证用户信息已经被正确保存、会自动登录且不会再次进入引导界面。

注：因篇幅有限，含所有步骤和测试分支更详细的完整代码覆盖测试并未展示在本文档内，此处只展示了一小部分代码覆盖测试内容。多环境测试和压力测试同样并未全部展示。

**多环境测试：**

在不同机型下运行本程序的情况如下：（调试时换几个机型/人，截一下图，列一下表）

**压力测试：**

1. 长文本日记测试，如图x所示。加载用时为x秒。
2. 大量图片日记测试，如图x所示。加载用时为x秒，全部图片顺利加载。
3. 大量景点管理页面测试，如图x所示。

（截图建议：把几十个(或如果有空的话…全部)景点全部划到我喜欢，然后全部设置为我去过，然后点击“我的景点”，测试加载时间；选择(看起来)最长的景点介绍，点开详细信息查看，测试加载时间，写一篇长约10000字(随便复制粘贴点什么)的日记，然后发布，展开查看，编辑，保存；写一篇包含10张以上图片和少量文本的日记，查看效果）

附录：参考资料

引用了什么第三方样式，是否需要写出来(你们详细看一下那两篇范文，如果要就在这里写)