**![](data:application/octet-stream;base64,)**

**本科生毕业论文（设计）**



**题 目** **基于wifi探针签到与node爬虫的导游助手系统的设计与实现**

**学 院 计算机学院**

**专 业 物联网工程**

**学生姓名 文红皓**

**学 号 2015141222070 年级 2015级**

**指导教师 黄戈**

**教务处制表**

**二Ο一九年四月十五日**

**摘 要**

物联网工程

学生 文红皓 指导老师 黄戈

**[摘要]** 随着中国人均消费能力的逐步提高，旅游现如今已成为当代人的一种主流消费方式，同时也成为了许多城市的支柱产业。旅行主要分为两种形式，自由行与跟团游。当代年轻人往往喜欢自由行，而中老年人出游，往往喜欢跟团游。对于一个旅行社而言，一次良好的出游，必须管理好团员，因为旅行社在带队过程中往往涉及比较麻烦的人员统计，现有方式一般是让导游做人工统计，这样效率较低，且容易出现纰漏。为了做好人员管理，越来越多的旅行社开始采用电子签到的方式，该文就是提出了一种新的电子签到人员管理方式。它基于wifi探针技术，在游客集合签到这个点上做一点尝试与突破，提高了签到效率，降低了可替代性，使签到地点更加灵活，同时，利用GPS定位和node爬虫为游客做了基于地理位置信息的定制化的旅游信息推送。该论文首先简要介绍了系统所使用的核心技术，包括wifi探针技术，node爬虫，react，微信小程序等技术的发展情况和应用场景，随后对项目本身做了需求设计，最后给出了具体实现情况和测试用例。该论文自主设计了一套可行的导游信息管理系统的设计方案，探讨了一种可行的新式人员管理系统建方案。

**[关键词]** WIFI探针；node爬虫；微信小程序;

**ABSTRACT**

Internet of Things Engineering

Student：Wen Honghao Adviser: Huang Ge

**[Abstract]** With the gradual increase in China's per capita spending power, tourism has now become a mainstream consumption method for contemporary people, and has also become a pillar industry in many cities. Travel is mainly divided into two forms, free tour and group tour. Compared with young people who often like to travel freely, middle-aged and elderly travel like to travel with the group. For a travel agency, a good travel must manage the members, because travel agencies have complex personnel management in the process of leading the team. The current method is generally to let the tour guide do manual statistics, which is less efficient and prone to appear mistakes. In order to do a good job in personnel management, more and more travel agencies have begun to adopt electronic sign-in. This article proposes a new electronic sign-in personnel management method. Based on the wifi probe technology, it makes a little attempt and breakthrough in the collection of tourists, which improves the efficiency of sign-in, reduces the substitutability, makes the check-in location more flexible, and at the same time, uses GPS positioning and node crawlers to push travel information for tourists. The paper first briefly introduces the core technologies used by the system, including wifi probe technology, node crawler, react, WeChat applet and other technologies development and application scenarios, and then design the project itself, and finally gave specific implementation and test cases. The paper independently designed a feasible design scheme of the guide information management system, and explored a feasible new personnel management system construction plan.

**[Key Words]** WiFi probe technology ; Node crawler ; Wechat applet

**目 录**

[1 绪 论 1](#_Toc8235241)

[1.1研究背景和研究意义 1](#_Toc8235242)

[1.2 国内外现状 2](#_Toc8235243)

[1.3 本文所做的主要工作 2](#_Toc8235244)

[2 开发工具介绍 4](#_Toc8235245)

[2.1 微信小程序介绍 4](#_Toc8235246)

[2.2 react框架介绍 4](#_Toc8235247)

[2.3 MONGODB介绍 5](#_Toc8235248)

[2.4 Express介绍 5](#_Toc8235249)

[2.5 Puppeter抓包框架介绍 5](#_Toc8235250)

[2.6 Echarts介绍 5](#_Toc8235251)

[3系统分析 6](#_Toc8235252)

[3.1 外部角色 6](#_Toc8235253)

[3.2 系统功能模块 6](#_Toc8235254)

[3.3 签到功能拓扑结构 7](#_Toc8235255)

[3.4 系统功能需求分析 8](#_Toc8235256)

[3.4.1用例图简介 8](#_Toc8235257)

[3.4.2全系统顶层用例图 8](#_Toc8235258)

[3.5 数据字典 10](#_Toc8235259)

[3.6小结 12](#_Toc8235260)

[4 系统概要设计 13](#_Toc8235261)

[4.1系统工作流程 13](#_Toc8235262)

[4. 2部分难点接口设计 14](#_Toc8235263)

[4.2.1用户绑定与自动登录接口 14](#_Toc8235264)

[4.2.2 游客wifi签到接口 15](#_Toc8235265)

[4.3系统开发环境 16](#_Toc8235266)

[4.4系统运行环境 16](#_Toc8235267)

[4.5小结 17](#_Toc8235268)

[5 系统的详细实现与设计 18](#_Toc8235269)

[5.1微信小程序 18](#_Toc8235270)

[5.1.1用户管理模块 18](#_Toc8235271)

[5.1.2 签到管理模块 20](#_Toc8235272)

[5.1.3游记日志 23](#_Toc8235273)

[5.1.4 通知 25](#_Toc8235274)

[5.2 web后端管理系统 26](#_Toc8235275)

[5.2.1 登录与注册模块 26](#_Toc8235276)

[5.2.2 导游信息管理 28](#_Toc8235277)

[5.2.3 行程管理 30](#_Toc8235278)

[5.3小结 36](#_Toc8235279)

[6 系统测试 37](#_Toc8235280)

[6.1 测试需求 37](#_Toc8235281)

[6.2测试目的 38](#_Toc8235282)

[6.3测试方法 39](#_Toc8235283)

[6.4测试范围 39](#_Toc8235284)

[6.5测试环境 39](#_Toc8235285)

[6.6 测试用例 40](#_Toc8235286)

[6.6.1微信小程序 40](#_Toc8235287)

[6.6.2 web后端管理系统 41](#_Toc8235288)

[**6.7** 测试结论 44](#_Toc8235289)

[6.8 小结 44](#_Toc8235290)

[7 总结与展望 45](#_Toc8235291)

[参考文献 46](#_Toc8235292)

[附录 47](#_Toc8235293)

[附录1：项目源代码说明 47](#_Toc8235294)

[致 谢 48](#_Toc8235295)

[声 明 49](#_Toc8235296)

1 绪 论

本章节主要描述该课题的研究背景，以及目前导游人员管理系统存在的必要性和目前市场上跟团游主要人员统计的方式。并在本章节末尾对本论文所做的工作进行了简要的描述。

## 1.1研究背景和研究意义

随着国内人均可支配收入的不断增加，第三产业占比的不断加重，旅游现如今已成社会中的一种主流消费方式。旅游主要可以分为两种，跟团游和自由行，自由行暂时按下不表，跟团游就是旅客自己到旅行社报到，旅行社安排游客的出行和日程，根据中国报告网的调查报告，2016年，跟团游的占比高达42.4%，且呈现上升趋势。与自由行不同，跟团游中很大一部分游客是中老年人，国内老龄化进程的加快使得跟团游市场不断扩大，以2016年末为节点计算，中国60岁及以上的人口占比高达16.7%，人数已经到达了2.3亿。与此同时，大约63%的老年人旅游时选择跟团游的形式[1]。可以看出，这一个市场是非常广阔的，一个体验良好的跟团游除了需要旅行社做好行程安排，另外非常重要的一点是需要妥善的统计好旅行过程中的人员出入情况，这样能提升旅行效率。现有旅行社方式使通过导游点名等人工方式统计，效率极其低下。针对这个现象，设计一个符合标准的，能提高签到效率的导游人员统计系统是非常必要的。

考虑到跟团游这种场景的特殊情况，这个系统需要三点要求：

1. 这个系统必须拥有很好的负载，能在同一时间段允许较多设备签到。
2. 签到过程必须是简单的，适合老年用户使用的。
3. 签到必须是不受地理因素制约的，可以随时随地发起。

基于上面描述的情况，在经过反复的思考后。决定使用wifi探针技术来做人员签到管理，前端实现以微信小程序为载体，辅以web后端管理系统来实现。

核心思路是使用wifi探针（wifi Probe）技术通过匹配wifi关键信息(ssid和mac码)来进行签到校验。所谓wifi探针技术是一种基于物理硬件，在特定范围内，可自动辨认移动终端设备（常见的有智能手机，笔记本电脑，平板电脑等）及其关键信息（一般是指mac地址，rssi值等）的技术[2]。需要注意的是，wifi探针一般不需要用户连接其wifi。此外，一般情况下，wifi探针无法取到用户的电话号，设备id等敏感数据。

所谓特定范围，通常是开放环境下的50米至200米，即wifi的可探测的范围。当然特定范围还和周围环境的具体情况有关系，比如当处于室内条件中，由于受到家具的遮挡和反射、其他电子产品产生的类似信号的干扰等因素的影响，一般范围会缩小。

基于以上原理，我们可以在不获取用户敏感信息的情况下对用户做精准定位，利用这个技术，能非常方便的统计人员出入来往情况，非常适合跟团游这种场景的人员管理。更重要的是它能完美符合前面提出的三点要求，wifi探针并不需要用户连接特定wifi，只需要检测该wifi的关键信息就可以了，这保证了系统拥有良好的负载且不受时间空间的制约（通过手机创建热点而不是固定wifi），使用基于微信开发的微信小程序能保证操作简单且覆盖面较广（由于微信小程序跨平台的特性）。

另外，为了增强微信小程序的工具属性，也为了提高小程序的交互能力，本系统在微信小程序里面设置了单独的旅游日志模块，用户可以点击该模块获取当地的旅行攻略，攻略信息来自马蜂窝和携程，通过node爬虫得到。

本项目探索了一种可商用的新型旅行团人员管理方式，通过wifi探针匹配周围存在的热点，以达到人员统计的目的，这种方式，既可以一定程度上避免虚假人员统计（代签，口头统计出错等）等现象等发生，另一方面可以提高人员统计的效率，这种方式不止可以使用在导游签到管理的场景下，还可以使用在其他需要灵活人员统计的管理场景下。

综上所述，本项目---基于wifi探针和node爬虫的旅行团人员管理工具的研究与实现是极为价值的。

## 1.2 国内外现状

现有跟团游旅行过程中的签到方式一般是点名签到，这样做的效率极其低下，把目光拓展到全行业，不止局限在旅游业，可以看到现在也出现了一些电子签到的方式。以GPS定位签到和wifi连接签到最为常见。

GPS定位签到是通过匹配用户地理位置来确定是否允许签到成功的，当用户的地理位置与服务器记录的位置误差小于允许的最大值时，就视为签到成功。这种方式较为灵活，不受时间空间约束。但是也有缺陷，即不同的智能设备定位是有差别的，这些差别可能会对系统造成较为明显的误差。

Wifi连接签到近年也流行开来，其思想是通过用户连接指定wifi来确定用户是否到场。但这也有着明显的缺陷，由于wifi连接在同一时间断需要连接较多的用户，因此wifi连接使用的wifi必须是特定wifi。因此这种方式在时间和空间的灵活性上有着较大的缺陷。

## 1.3 本文所做的主要工作

本论文主要讲述的是设计并且实现一个基于wifi探针和node爬虫的导游辅助系统的设计和实现过程。论文正文分为七个章节：

第一章：绪论，介绍论文的背景意义，业界现状以及该论文的文章结构。

第二章：对系统设计时涉及的关键的技术，微信小程序，echarts，node，react框架等相关技术的介绍。

第三章：系统分析项目的目标用户。简述系统用例图，将系统功能按照模块划分，最后解析了数据库表结构设计。

第四章：系统概要设计，对系统的工作流程进行了讲解，给出了部分接口的设计逻辑。最后简单描述了系统开发环境与运行环境。

第五章：系统具体实现，给出了实现的主要界面，对各个模块进行了详细讲解，对于部分较为复杂的功能，给出了实现代码。

第六章：系统测试，给出了具体测试用例和结论。

第七章：总结与展望。

# 2 开发工具介绍

## 2.1 微信小程序介绍

在微信小程序官网中，我们可以知道，小程序作为一种新的依托微信平台的程序形式，它能帮助开发者快速地完成一个跨平台的软件。依托微信九亿用户，它在微信内可以被快速地获取和传播。相比于传统h5应用，由于依托了微信内部很多组件，微信小程序具有更好的用户体验。

小程序的开发一般是用微信提供的微信开发者工具开发，与一般前端开发三件套html，css，js不同，它用了wxml, wxss，js作为三件套，前两者是微信小程序特有的语言[3]。但是需要注意一点的是，微信小程序的js并不是运行在浏览器或者node环境中，而是微信的上下文环境，因此并不能够直接进行dom操作或者使用node相关输入输出的api，另外需要注意的一点是微信小程序运行必须要将服务器运行在具有可信任证书的https服务器上，且端口必须为443或者80。

## 2.2 react框架介绍

作为facebook的明星开源项目，它最开始是facebook为了架构instagram所设计的一套MVC框架，和其他MVC框架都有不同之处。它拥有着极其出众的性能，设计思想别出心裁，代码的实现逻辑却非常简单。因此，自2015年起，react受前端工程师的关注程度越来越高，甚至在github上它已经成为了star数目最多的项目之一。这标志着它已经是前端领域最受欢迎的技术之一。

React与传统前端开发框架最大的差别是引入了虚拟dom的思想，我们都知道浏览器最耗费性能的一步是重绘[4]，虚拟dom正好可以减轻重绘带来的性能影响：简单的来说它在浏览器端使用js实现了一套diff算法，通过js模拟dom结构，然后比较dom结构差异，找出差异并反馈到真实的dom上去。换句话说，react的dom操作并不是通过操作真实dom结构进行的，每当数据变化（调用setstate）时，React都会通过内置的diff算法重新模拟整个DOM结构，然后将比较这一次的DOM结构和上一次的DOM结构的差异，得到DOM结构的区别，接着只是把发生了变化的部分进行真实DOM的渲染，以此节省浏览器开销[5]。需要注意的一点是dom渲染并不是同步的，而是异步的，这样带来的好处是react的dom更新是集中在一处的，react将批量处理更新而不是每次都处理。虚拟dom的引入使得我们在前端开发过程中只需要注意数据的变化下，而不需要关心界面是如何渲染的，这极大的提高了前端开发的效率。

另外，虚拟dom不止提高了性能，还使组件化成为了可能：所谓组件化，就是ui组件的封装与复用，封装就是将ui组件的数据，逻辑，视图进行封装，对于外界而言，组件就是一个黑盒子，只要输入对应的数据就能够展示出来。

## 2.3 MONGODB介绍

介绍mongodb之前，先要了解一个概念，nosql，“Not Only Sql”，顾名思义是指“不仅仅是SQL”，也是说非关系型数据库。非关系型数据库很早就有人提出，但是直到近年才渐渐流行开来。从2009年起，非关系型数据库的走势越来越高涨。NoSQL的关键在于使用非关系型的数据存储。Nosql主要在读写，成本方面相比于传统的关系型数据库，有着非常明显的优势，但同时它也缺乏对sql的支持[6]。

作为一种常见的nosql数据库， 用c++编写的MongoDB在业界有着非常广泛的应用。腾讯，阿里等厂商都在自家的产品中使用过MongoDB。它在性能和使用部署方面有着非常明显的优势。也因此。mongodb比较适合网站数据的实时插入更新，往往用于高伸缩性的场景。

## 2.4 Express介绍

Express 是一个基于node的服务端框架，它仅仅提供最基础的应用，但是它有着丰富的中间件可以使用，这些中间件将帮助它构建多种多样，适应不同场景和需求的模块[7]。它有着卓越的健壮性，能够在极短的时间内完成一个拥有丰富功能的网站。express作为一种简洁有效的服务端构建工具，它使js在服务端领域大放异彩，而摆脱了浏览器的一亩三分地。

## 2.5 Puppeter抓包框架介绍

google开源的node抓包工具Puppeteer支持调用非常丰富的基于chrome的api。Puppeteer的英文意思是操纵木偶的人，这是指它可以在内存中模拟V8引擎的chrome浏览器，这使得它操作dom非常的方便[8]。因此相较于传统爬虫框架PhantomJs或者Selenium，它有着更好的用户体验。此外，传统的爬虫在爬取动态页面，懒加载页面有着很大的局限性，另外也时常无法模拟浏览器做一些点击，翻页等操作，puppeteer解决了这个问题，通过模拟浏览器，使得能够爬去更完整，更有效，更多样的信息。

## 2.6 Echarts介绍

ECharts是一个使用JavaScript实现的数据可视化库，由百度前端部门开源，广泛应用在百度，腾讯，阿里等公司的项目中。它能够流畅的运行在各种web环境中，Echarts具有十分卓越的兼容性，兼容IE8/9/10/11，Chrome，Firefox，Safari等当前绝大部分浏览器，它的底层是 ZRender这个轻量级的矢量图形库，能够高度定制化的数据可视化图表[9]。

# 3系统分析

## 3.1 外部角色

此项目外部角色有管理员(导游)，普通用户(游客)两部分。

管理员：管理员指的是导游，他们在web和小程序端都有权限，在web端，管理员主要是管理用户和旅程，负责普通用户信息管理，旅程信息的管理查看，以及一些基础信息的统计;在小程序端，管理员主要有wifi热点信息的发布管理以及通知信息的发布两种权限，除此之外，考虑到普通用户可能没有携带手机，还提供了代签的权限，用于与普通用户交互。

普通用户：普通用户在本项目中指的是游客，普通用户只拥有小程序的权限，没有登录web后台关系系统的权限。 在微信小程序端，系统为游客提供了签到(基于wifi探针)，查看同城游记(基于node爬虫)，通知信息浏览，旅程信息浏览的功能。 核心功能是基于wifi热点的签到，小程序会通过匹配周围是否存在于导游构建的热点关键信息(ssid和mac码)相同的热点，如果存在，视为签到成功。

## 3.2 系统功能模块

基于wifi探针和node爬虫的导游助手系统的设计与实现项目主要由两部分组成，微信小程序和web后台管理系统。微信小程序提供给所有用户（导游和游客）使用，而web后端管理系统只提供给管理员（导游）使用。

按照功能模块划分，微信小程序具有用户账号管理，签到管理，通知管理，旅游日志管理这四个模块。Web后端管理系统拥有账号管理，用户信息管理，旅程信息管理这三个模块，每个模块又有着对应的不同功能，具体划分见图3-1。

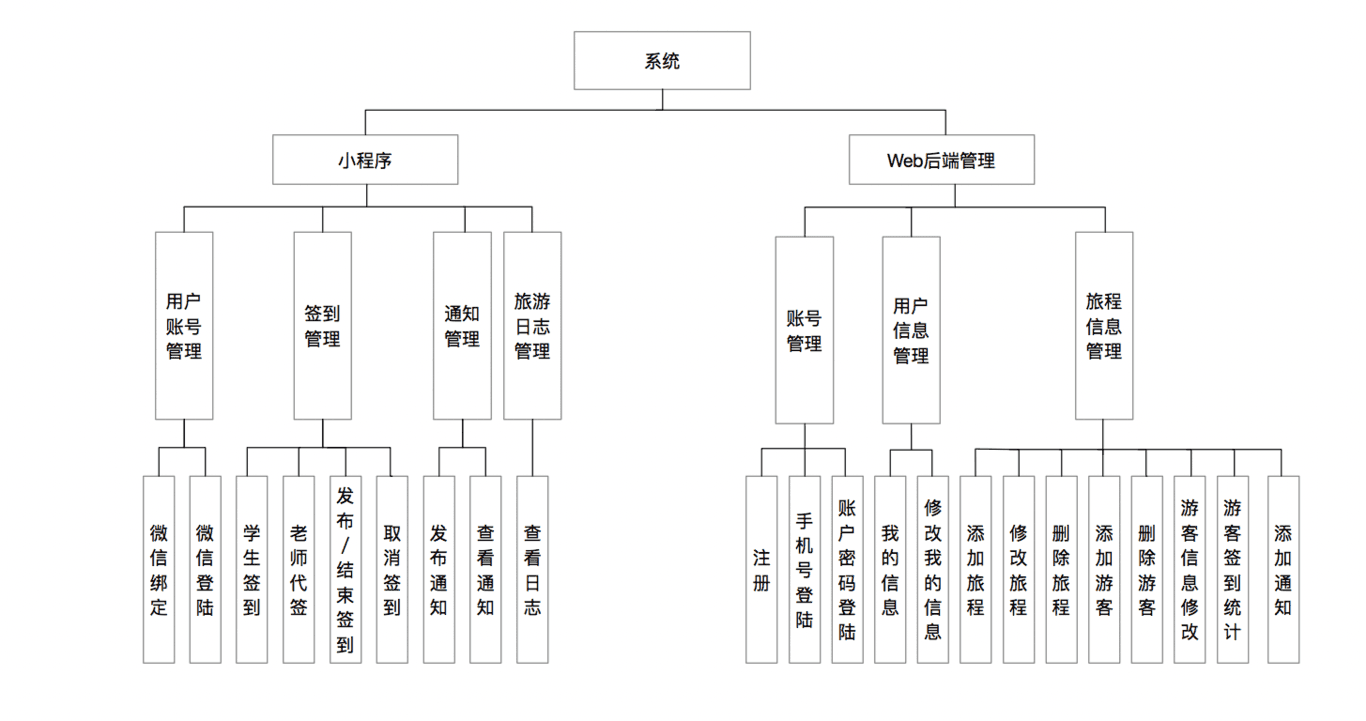
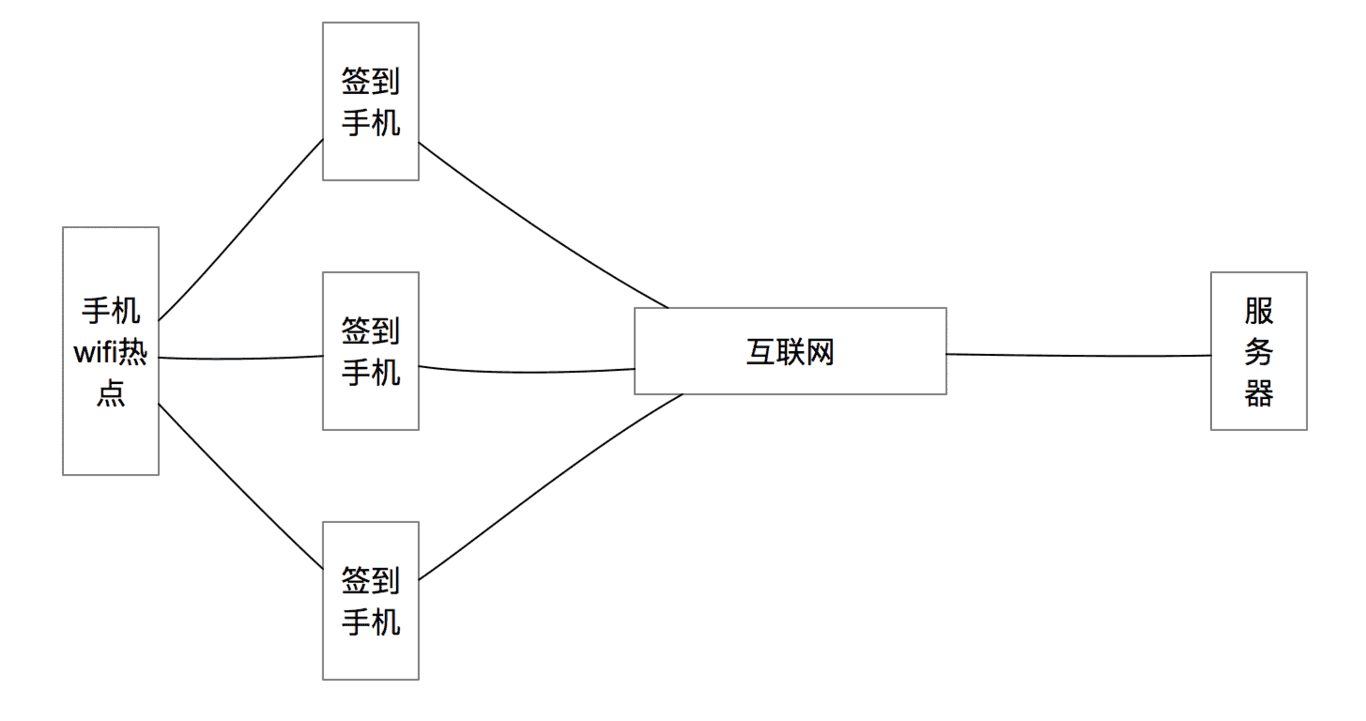


图3-1 系统功能模块图

## 3.3 签到功能拓扑结构

该项目是基于wifi热点进行人员统计，因此签到系统主要由wifi热点（导游自主创建或者其他热点），签到端（游客手机）和服务器组成。手机wifi热点提供wifi的名称（ssid）和mac地址，并将它们上传到服务器。签到端通过检测周围存在的wifi，匹配服务器中的地址来进行签到，并随后将匹配结果返回给服务器储存。这一功能的系统拓扑图如下3-2。

图3-2 签到拓扑图

## 3.4 系统功能需求分析

### 3.4.1用例图简介

用例图是由角色，用例以及它们之间关系构成的描述用户与系统的交互关系的逻辑视图[11]，其要素有三个：角色，用例，用例图。用例图主要可以用于系统，子系统或者功能建模，它是从外部观测得到的系统功能模型图。

### 3.4.2全系统顶层用例图

系统顶层用例图包括了导游信息管理，游客信息管理，旅程管理，wifi签到管理，通知信息管理五个大方向上的模块。分散在后端管理web系统和微信小程序两块上，小程序顶层用例图如图3-3，web后端管理系统顶层用例图如图3-4。

![](data:application/octet-stream;base64,)

图3-3 小程序的顶层用例图

![](data:application/octet-stream;base64,)

图3-4 web的顶层用例图

## 3.5 数据字典

数据字典就是对于本项目中涉及到的数据的名称，在代码中使用的变量名称以及变量的类型，还有对于该数据的说明。数据字典的目的就是使用户能够详细了解各个不同的数据在数据流程图中的作用。下图为本项目的数据字典。

用户信息表数据结构定义如表3-1所示:

表3-1系统相关数据元素的定义-用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **数据元素名** | **数据类型** | **取值范围** | **使用说明** |
| 用户名 | userName | string | 字符、数字 | 用户的登录名 |
| 密码 | password | string | 字符、数字 | 用户的登录密码 |
| 用户id | userId | string | 数字 | 用户id |
| 用户状态 | status | string | G或者T | G表示导游，T表示游客 |
| 用户头像 | userImg | string | 字符，url | 用户头像地址 |
| 用户邮箱 | email | string | 字符、数字 | 用户邮箱 |
| 用户电话 | userPhone | string | 数字 | 用户电话 |
| 用户公司 | company | string | 字符 | 用户公司 |
| 用户职业 | userJob | string | 字符 | 选填，只有游客才需要此字段，表示用户职业 |
| 自我介绍 | introduce | string | 字符 | 自我介绍 |
| 旅程信息 | guideList | Array | 数组，每一项都是一个对象 | 导游正在参与或者参与过的旅程或者游客正在参与或者参与过的旅程 |

续表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 旅程id | guideId | string | 数字，  guideList每一项都拥有的属性 | 旅程id和guideSN构成旅程唯一标志符号 |
| 旅程序号 | guideSN | string | 数字，guideList每一项都拥有此熟悉 | 旅程序号，与guideId构成旅程的唯一标识符 |

旅程信息表数据结构定义如下表3-2所示：

表3-2数据结构定义表-旅程信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **数据元素名** | **数据类型** | **取值范围** | **使用说明** |
| 旅程号 | guideId | string | 数字 | 旅程id |
| 旅程序号 | guideSN | string | 数字 | 旅程序号 |
| 用户id | guideName | string | 数字，字符 | 旅程名 |
| 导游号 | guiderId | string | 数字 | 同userId，导游唯一标识符 |
| Wifi ssid | guideSSID | string | 字符 | Wifi ssid，用于签到 |
| Wifi mac | guideMac | string | 字符 | Wifi mac码 |
| 旅程图片 | guideImg | string | 旅程图片 | 旅程图片 |
| 签到状态 | status | string | 字符，数字 | 当前签到状态 |
| 旅程信息 | guideInfo | string | 旅程信息 | 当前旅程信息 |
| 游客 | tourists | Array | 数组 | 表3-3tourist说明表详细说明 |
| 通知 | notice | Array | 数组 | 表3-4notice表详细说明 |

表3-3 toutist说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 数据元素名 | 数据类型 | 取值范围 | 数据说明 |
| 游客id | touristId | string | 数字 | 游客唯一标志符 |
| 游客状态 | touristState | String | 1或者0 | 0表示已经签到了，1表示未签到 |
| 签到状态 | signInCount | Array | 对象 | 每个值，都是一个对象，包含两个属性，tag含义是这是那一次签到，isSign  含义为是否签到 |

表3-4 notice说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 数据元素名 | 数据类型 | 取值范围 | 数据说明 |
| 标题 | title | string | 字符 | 通知标题 |
| 内容 | content | String | 字符 | 通知内容 |
| 时间 | time | Array | 字符 | 通知创建时间 |

## 3.6小结

本章对系统进行了系统分析。

3.1节介绍本项目涉及到的两个外部角色，一个是管理员(导游)，一个是普通用户(游客)，将他们分web端和小程序端进行了介绍。

3.2节和3.3节按照小程序和web两端将系统进行模块划分，并介绍了wifi探针签到的拓扑图。

3.4节介绍了本项目的全系统顶层用例图。

4 系统概要设计

## 4.1系统工作流程

本系统是一个基于wifi探针的，目标人群是导游的旅程人员管理系统，分为web后端管理系统和微信小程序。web后端管理系统只有管理员拥有权限，给管理员提供了注册用户，游客管理，旅程管理，通知管理等功能。小程序端，游客和导游都拥有权限，导游拥有创建wifi热点签到信息，代签，发布通知，查看旅程详情的权限。游客拥有签到，查看本市旅游日志，查看通知，查看旅程详情的权限。下面是系统的工作流程：

小程序端:

1. 初次进入小程序，会要求绑定微信，游输入导游账号关联微信账户，游客输入导游给的账号关联微信账户，随后系统将自动登录。
2. 自动登录后，进入主界面，底部是切换栏，除了主界面外，随着点击的不同，可以切换到旅行日志和通知消息栏目。导游的主界面是自己正在跟的旅行团，游客的主界面是当前参与的旅程。
3. 导游点击指定的旅程，进入旅行团列表，在这里可以看到旅程的详细信息，以及每个旅行团成员，在这个页面，导游应该拥有开始签到，终止签到，代签，发布通知的权限，并且能够看到每个游客的签到情况。
4. 游客可以直接在主界面看到当前旅程是否开始签到，并且如果已经开始签到，能够点击签到按钮更新签到状态。当游客点击签到后，如果游客周围存在导游所上传的热点的关键信息，则视为签到成功，否则签到失败。
5. 游客和导游点击切换栏中的游记栏目时能够看到关于当前城市旅行日志和攻略，数据来源于马蜂窝和携程。
6. 游客和导游点击通知时，可以看到每一个旅程的通知详情(包括时间，标题和内容)

Web后台管理系统：

1. Web后端管理系统具有注册和登录功能，注册和登录应仅开放给导游使用，注册时需要采集导游关键信息;登录时应提供密码登录和手机号登录，忘记密码应该允许导游找回。

2. 登录成功后，用户可以看到自己的信息，左侧应提供导航栏，供导游切换，导航栏应提供切换到个人信息管理，旅程管理。

3. 个人信息管理允许导游更新自己的信息。

4. 旅程管理允许导游修改，创建，删除旅程，同时对于每个旅程，能够对它的游客信息进行修改，增加，删除和签到情况统计（图形化）。此外，旅程管理能够允许用户发布删除通知。

## 4. 2部分难点接口设计

本系统由两部分组成，微信小程序和web后端管理。

微信小程序分为导游端和游客端，具有登录绑定，签到管理，旅行日志查看，通知查看四个模块，web后端管理系统则有登录注册管理，用户信息管理和旅程管理三大模块，这其中涉及很多接口，选取部分难点接口进行说明:

### 4.2.1用户绑定与自动登录接口

**描述：**用户绑定与登录接口是用于小程序的接口，其作用是初次登录时，用户需要用web后端管理系统提供的账号绑定微信，非初次登录时，系统将自动登录。

**实现逻辑：**自动登录与绑定的实现逻辑，借用了微信小程序提供的官方接口login，小程序初始化时，首先调用wx.login;这个接口来判断是自动登录还是需要绑定，这个接口会返回一个code码，code码是微信提供的当前用户临时识别码，具有5mins的有效期，将这个code码发送到后端，后端使用code码，连同小程序的appid和appsercert(这两者在注册小程序时微信提供)按照微信的要求发送到指定的地址，这个地址将返回一个openid，这个openid是该小程序此用户的唯一识别码，数据库可以通过监测是否存在这个openid判断此微信用户在系统内是否存在对应账户，如果存在，返回用户数据，前端渲染主界面，如果不存在，返回字符串“1”，前端提示用户绑定账户。绑定逻辑类似，前端将code，用户id和密码返回给服务器，服务器通过code，appid和sercret得到用户openid，数据库通过前端传来的用户id和密码找到对应的用户，给该用户储存openid，这样下次登录时直接通过openid就能够完成登录。

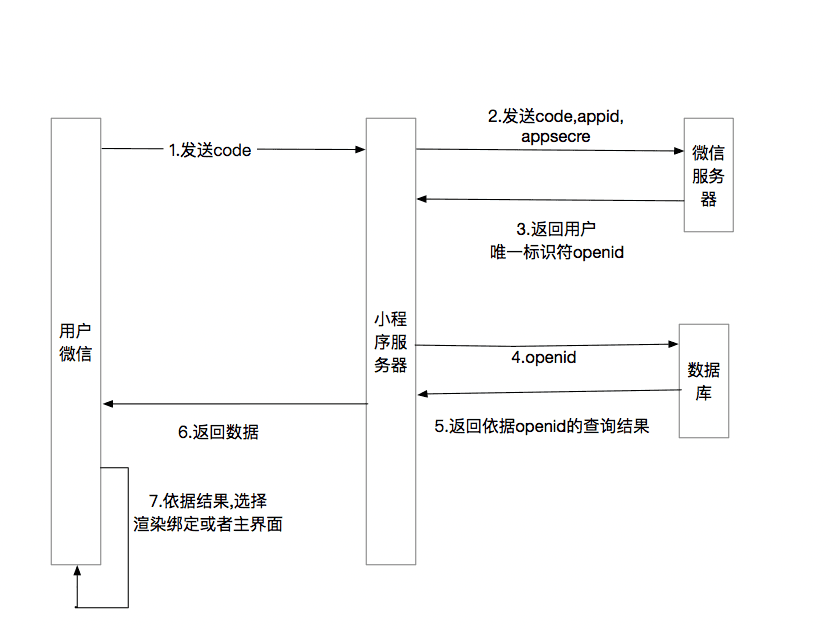
账户绑定逻辑图如图4-1所示：

图4-1 账户绑定逻辑图

自动登录逻辑图如下图4-2所示：

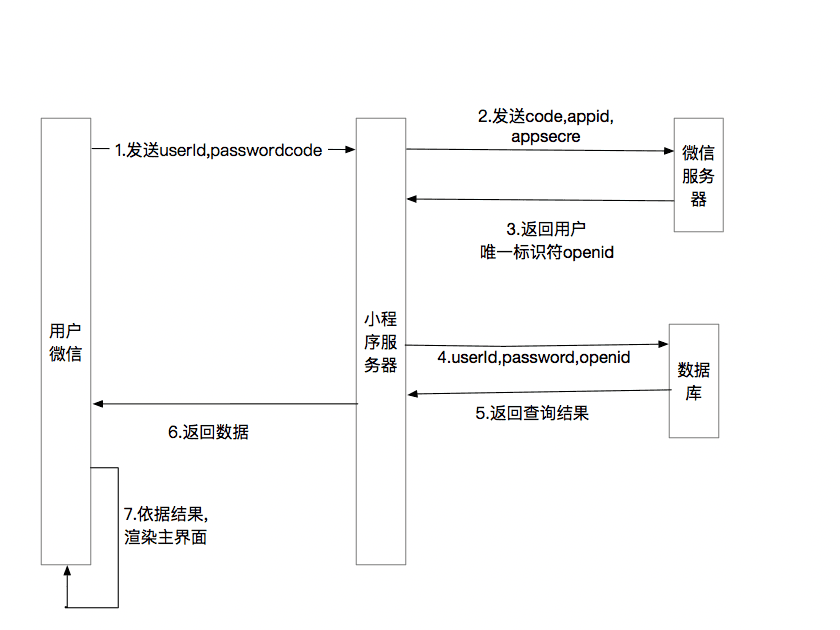


图4-2 自动登录逻辑图

### 4.2.2 游客wifi签到接口

**描述：**签到接口主要分为两种，一种是游客wifi签到接口，一种是导游代签接口，游客wifi签到接口是通过小程序识别周围的wifi情况，搜索是否存在和导游上传的wifi关键点匹配(wifi名称和mac码)的信息实现的;导游代签接口则很简单，只要导游核对游客信息无误(头像，学号)，就可以进行签到。本节所阐述的签到接口是指游客wifi签到接口。

**实现逻辑：**

导游端：由于微信小程序不支持自动打开热点，因此首先需要导游手动打开热点，随后导游点击小程序内开始签到按钮。此时小程序应该提示导游输入热点名称。在导游输入了正确的热点名称后，小程序监测周围是否存在该名称的热点，如果存在这个热点，将此热点的mac地址和wifi名称一起上传到服务器。随后小程序通知导游准备工作已完成，可以允许游客进行签到了。

游客端: 收到导游允许签到的通知后，游客可点击签到按钮，如果周围存在名称和mac地址都和导游上传信息相同的热点，视为签到成功。

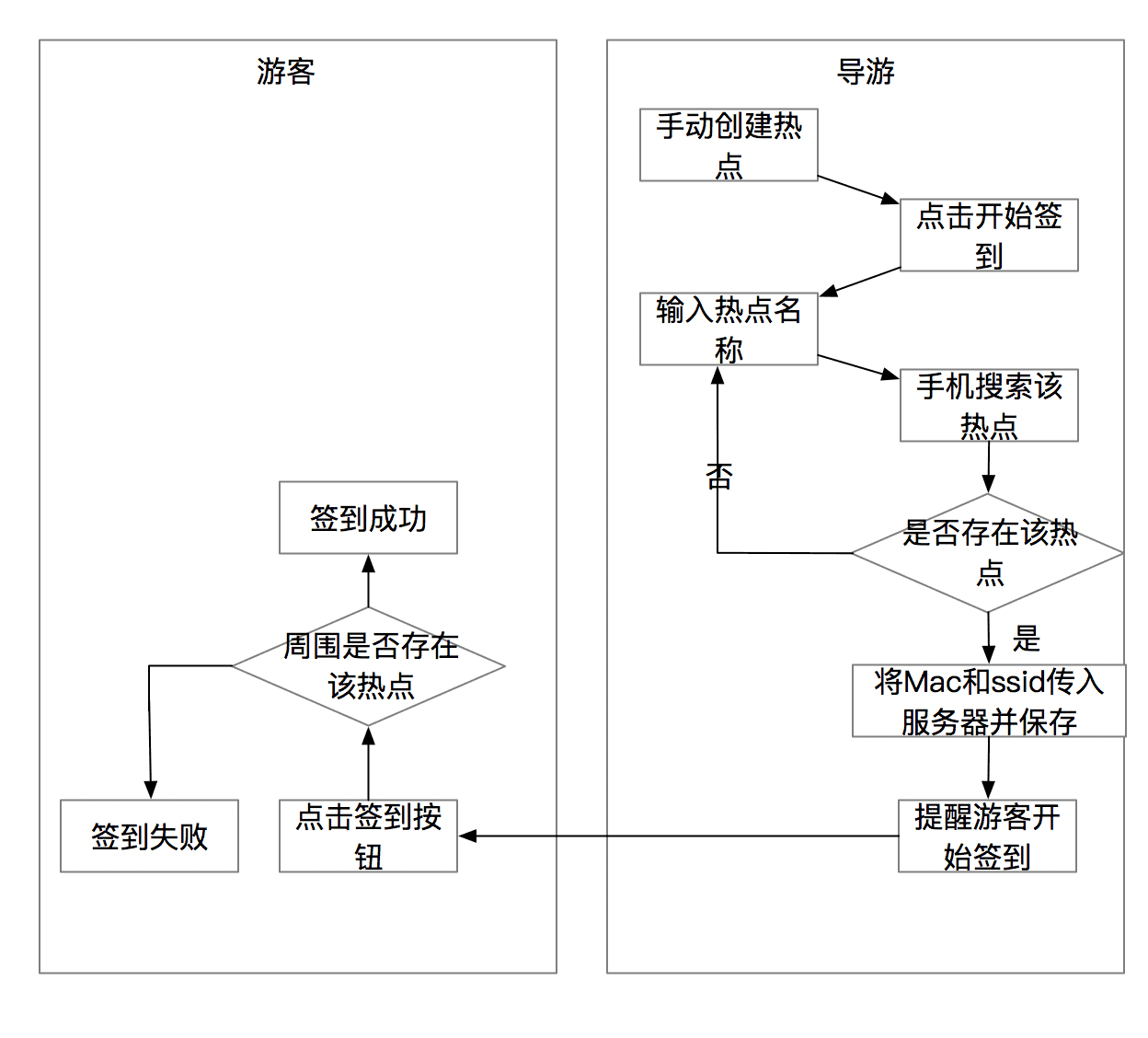
Wifi签到逻辑图如图4-3所示：

图4-3 游客签到接口设计图

## 4.3系统开发环境

开发环境：mac os 10.14

服务器环境：centos 7

开发语言：js，html，css

## 4.4系统运行环境

小程序

手机：运行微信5.3及以上版本的android(要求android5.0以上)或者ios(要求ios10以上)设备。 需要注意的一点是部分手机可能不支持getWifiList这个api，因此在部分手机上是无法使用这个小程序的。

Web端

Chrome，firefox，safria或者ie8+即可

## 4.5小结

本章主要是系统的概要设计。

4.1节介绍了系统的工作流程，描述了微信小程序和web后端管理系统的工作流程

4.2节主要是对部分接口的实现进行了设计，给出了具体的逻辑和实现方案。

4.3节和4.4节主要是阐述了小程序的开发环境和运行环境。

# 5 系统的详细实现与设计

本章节主要从两个个维度(小程序和web后端管理)来描述系统的实现情况与设计思路。小程序端包括用户账号管理，签到管理，通知模块和旅行日志这四大模块；web后端管理系统主要是登录注册模块，用户信息管理，旅程管理这三大模块。两个维度，总计七大模块合一，最终组成了一个完整的系统。本章将展示这几大模块的具体实现结果，部分难点会阐述实现逻辑和代码。

## 5.1微信小程序

### 5.1.1用户管理模块

用户管理模块在微信小程序上的体现主要是用户账号绑定，初次登录时，用户需要用web后端管理系统提供的账号绑定微信，非初次登录时，系统将自动登录，绑定实现结果见下图（图5-1）。



图5-1 用户绑定图

**描述：**

如上图5-1所示，用户初次登录，会提示用户输入用户id和密码进行账号绑定，用户id和账号需用通过web后端管理网站注册得到，小程序端并没有开放注册，非初次登录，会监测当前的微信账号做自动登录。

关键代码:

绑定接口:

function wxBind(){

let params={

userId:req.body.userId, //用户id

password:req.body.password, //绑定的密码

code:req.body.code, //微信传过来的code码

appid:wx.wx.appid, //小程序appid

appsercret:wx.wx.appsercret //小程序密钥

}

let openid=getOpenId(params.code,params.appid,params.appsercret)//获取用户openid

let user=getUser(wx.userId,wx.password)//查找用户并返回对应的用户信息

if(user===undefined){fail()}//如果没有这个用户，错误处理

else{

setOpenid(user,openid)//将openid存储到对应用户中去

}

}

登录接口:

function wxloginIn(){

const params={

code:req.body.code,//微信返回到用户code

appid:wx.wx.appid,//小程序appid

appsercret:wx.wx.appsercret//小程序迷药

}

let openid=getOpenid(wx.code,wx.appid,wx.appsercret)//向微信服务器请求用户唯一标志符

let user=getUser(openid)//根据openid请求用户信息

if(user){

res(user)//向客户端返回数据

}else{

fail()//错误处理

}

}

### 5.1.2 签到管理模块

**5.1.2.1 首页**

用户在登录后，进入首页就是签到管理模块，这里列出了用户参与过和正在参与的所有旅程。相关实现结果见图5-2和图5-3.

 ![](data:application/octet-stream;base64,)

图5-2 首页 5-3 首页-旅程详情

**描述：**

如上图5-2所示，用户在登录完成后，会进入首页，首页中部是旅程列表，最下方是切换栏。点击切换栏的我的，游记，通知分别可以进入首页，游记中心，通知中心。旅程列表的每一项都有详情和签到两个按钮，点击详情可以看到旅程详情(如图5-3)，而点击签到，则依据用户身份不同，执行的功能也不同，如过是导游点击签到，会进入签到详情页(见5.1.2.2描述)，如果是游客，会直接执行签到。

游客执行签到的逻辑见4.2.2描述，简而言之是通过匹配当前地理位置的wifi信息，找寻是否存在老师上传的wifi关键信息来确定是否进行签到。 如果签到成功，签到按钮下方“正在签到”会变成“完成签到”字样。

实现代码：

function signIn(){

let ssid=server.ssid;//服务器储存的wifi的ssid

let mac=server.mac;//服务器储存的wifi的mac地址

let list=getWifiList()//得到当前用户周围的wifi信息

let tag=fail;//签到状态，默认失败

list.forEach(item=>{

if(item.ssid===ssid&&item.mac===mac){

tag=true//如果存在wifi匹配，将状态修改为签到成功

}

})

setSignInTag(tag);//向服务器中存储状态

}

**5.1.2.2 游客列表页**

这是导游特有的页面，从首页点击对应的旅程，进入此页，这个页面具有很多功能，包括控制是否进行签到，发布通知，查看学生的签到情况或者具体信息，代签，取消签到等功能。实现结果见图5-4和5-5。

![](data:application/octet-stream;base64,)![](data:application/octet-stream;base64,)

图5-4 游客列表页 5-5 游客列表页-发布通知

**描述：**

如上图5-4所示，导游点击对应的旅行后，进入游客列表页。游客列表页的上方是该旅程的部分信息，包括旅程号，序号，旅行团总人数，出发地点以及签到情况；最下方是两个蓝色的按钮，左边是开始签到/结束签到按钮，右边是添加通知按钮；中间是游客列表，每个项由头像，游客名字，详情按钮，签到/取消签到，签到状态五个部分组成。点击详情按钮，可以查看该游客的详细信息，点击签到按钮，可以帮助游客代签(以防游客手机故障吗，不会操作等意外导致游客本身无法完成签到的情况)。点击添加通知，跳出添加通知的框(图5-5)，导游可输入通知名称和详情随后点击提交更新通知，加的通知可在通知管理处看到，详情见论文5.1.4描述；点击结束签到按钮，结束本次签到，结束签到按钮状态会变为开始签到，点击开始签到，将允许游客和导游签到，具体实现逻辑见论文4.2.2描述。

关键代码：

开始签到：

function startSignIn(){

//由于小程序不支持创建热点，因此首先需要手动创建热点

let ssid=wifiInput();//获取前端输入的wifi名称

let list=getWifiList();//获取周围存在的wifi列表

let mac='';//代上传的mac地址

let tag=false;//这个tag指周围是否存在上面定义的名称为ssid的wifi。默认不存在

list.forEach(item=>{

if(item.ssid===ssid){

mac=item.mac;

tag=true;

}

})

if(tag){

uploadServer(ssid,mac);//将这个ssid和mac存储到服务器中

}else{

noWifi()//没有指定wifi的情况

}

}

### 5.1.3游记日志

从首页点击切换栏的游记，进入游记首页，在这里可以看到用户所在城市的热门景点游记。从首页中点击具体的攻略进入游记详情, 在这里可以看到游记具体内容。

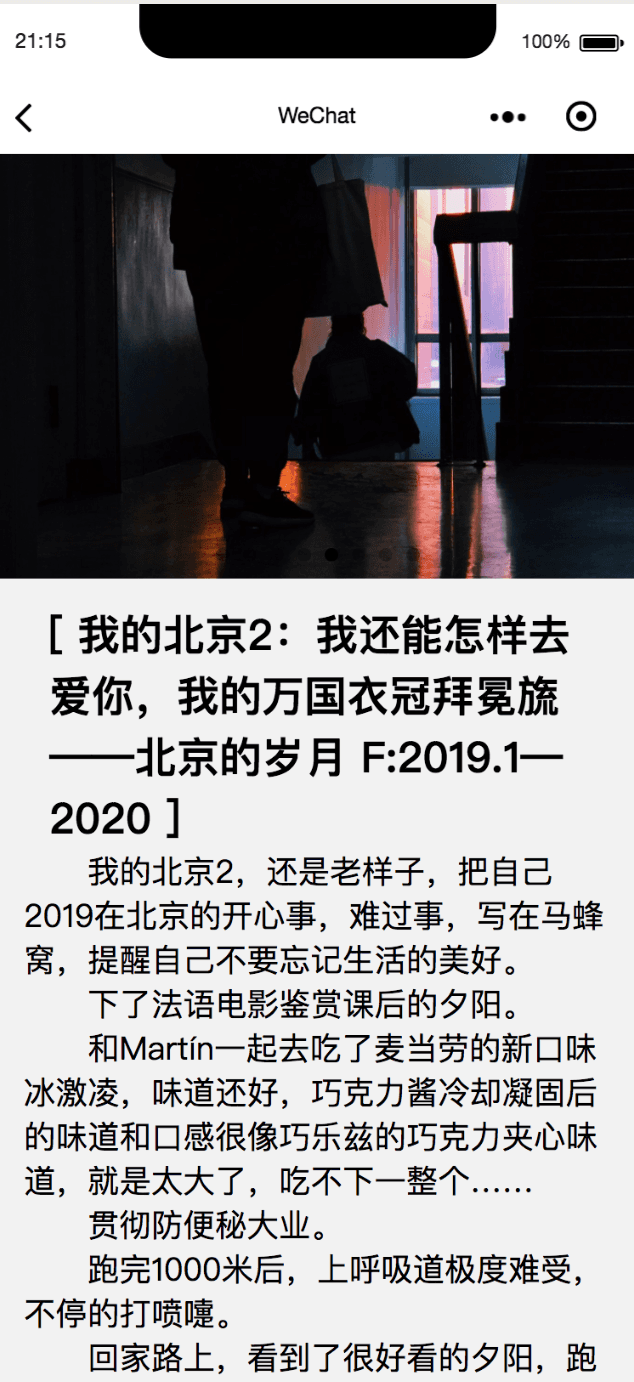


图5-6 游记首页 图5-7 游记详情

**描述：**

如上图5-6 所示，这是游记首页，用户从首页点击游记tab，进入此页，在这里可以看到本城市的旅游攻略，点击任意一个你感兴趣的攻略，将进入攻略详情页（例如图5-7），在这里将看到攻略的具体内容。游记首页的内容是通过爬虫马蜂窝获取的。

关键代码：

服务端爬虫：

function getData(){

//首先需要客户端通过getLocation你解析得到定位城市

let city=req.body.city;//获取客户端传过来的城市

let html=getUrl(city);//得到马蜂窝或者携程指定城市页面的html文档

let data=filter(html);//对这个html做正则匹配；返回的数据是一个数组

return data

}

### 5.1.4 通知

从首页点击切换栏的通知，进入通知首页，在这里可以看到用户当前所有的旅程 ，点击任意一个旅程，进入通知详情页，在这里就能看到通知详细信息。

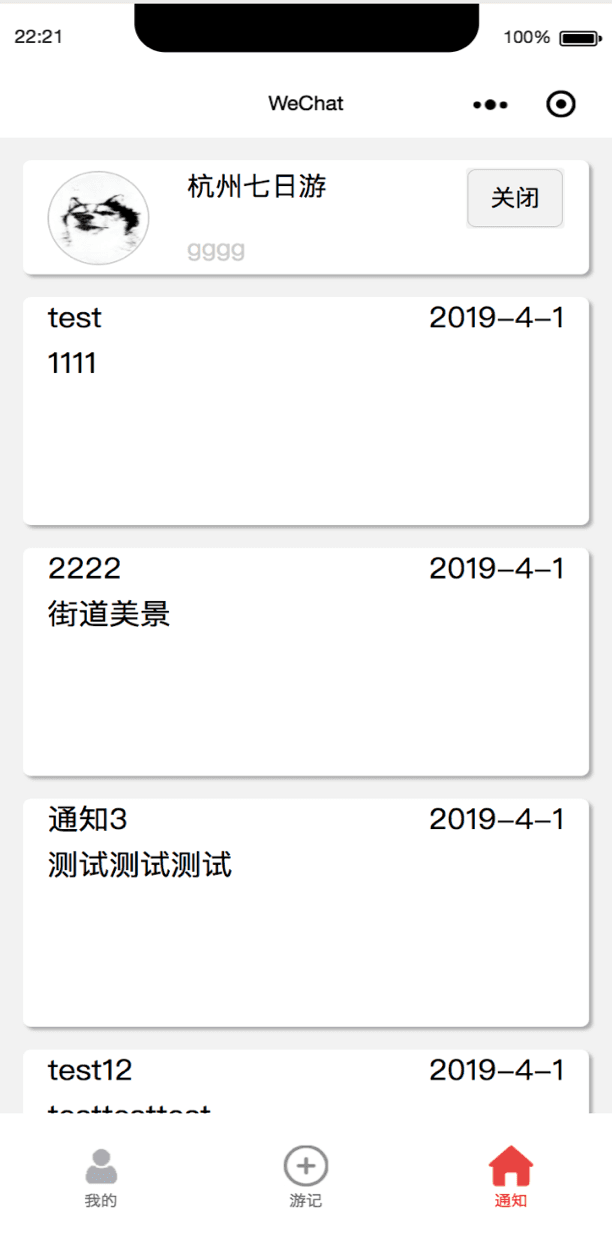
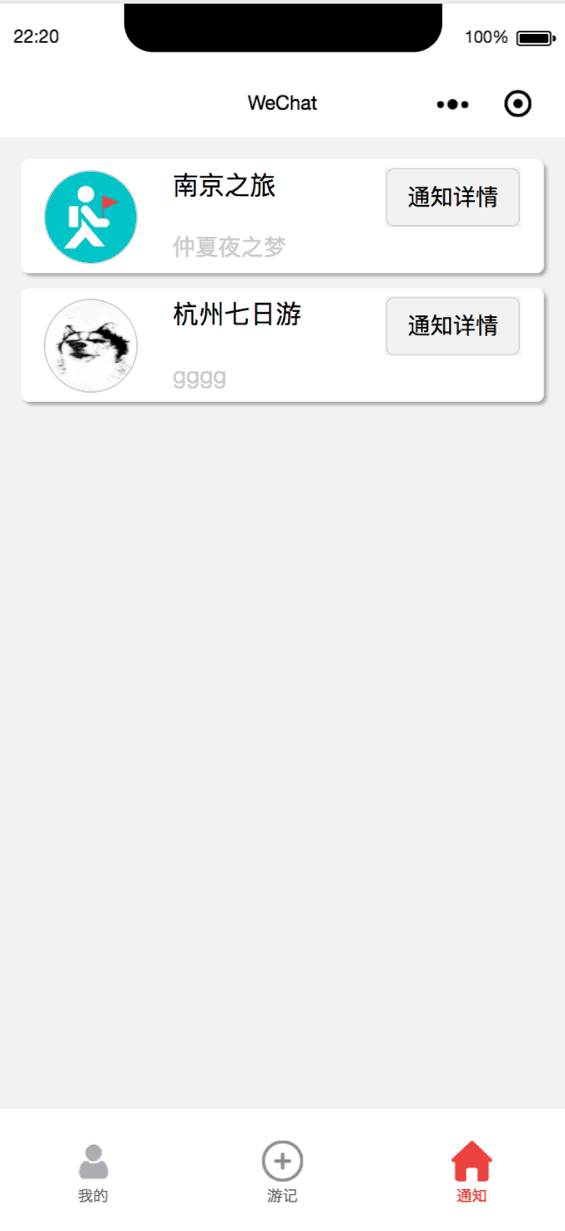


图5-8 通知首页 图5-9 通知详情

**描述：**

如上图5-8，这是通知首页，用户从首页点击通知tab，进入此页，在这里可以看到该用户参与的所有旅程，点击对应的旅程进入通知详情页（如图5-9），在这里能看到通知详细内容，通知一般是导游发布的，包含发布时间，标题，内容三个部分。

## 5.2 web后端管理系统

### 5.2.1 登录与注册模块

初次登录web后台管理模块，会先进入登录模块，如果用户没有账号，需要事先注册。 登录可以通过用户名与密码或者手机验证码登录

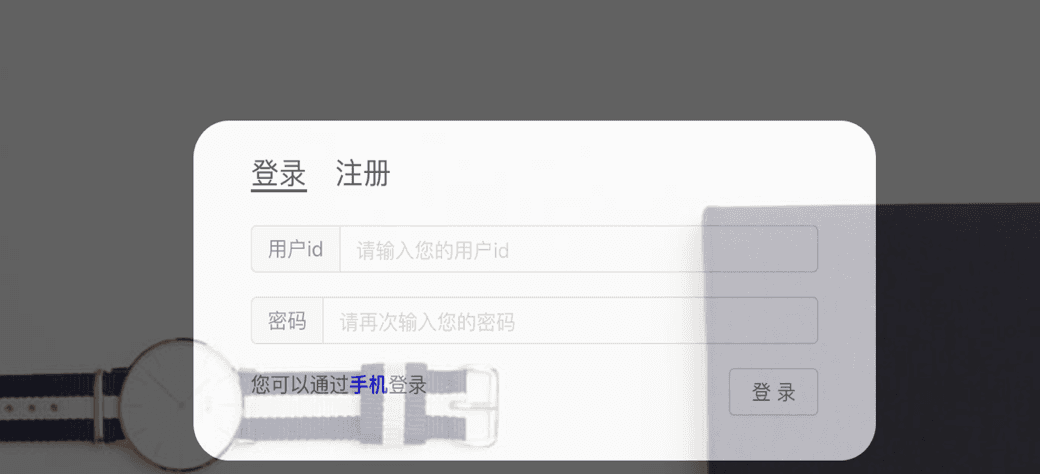


图5-10 普通登录



图5-11 手机验证码登录



图5-12 注册图1



图5-13 注册图2

**描述：**

如上图5-10和5-11 . 这是登录的可选项，登录允许账号登录和手机验证码登录。图5-12是注册栏，用户需要填写自己的用户id，姓名，手机号，邮箱，密码，并且上传头像。填写完成后，点击确定， 随后系统会向注册的手机号发送验证码（如图5-13），填写收到的验证码，如果填写无误，系统将核验填写信息是否符合以下标准：

1.Id没有与数据库中已有账号重复。

2.手机号没有与已有账号重复

3.密码与重复密码相同，且位数大于6位，小于12位。

符合以上标准系统将自动注册并登录账号，web将跳转到系统首页。

### 5.2.2 导游信息管理

**5.2.2.1 我的信息**

用户在登录后，进入首页就是导游信息管理模块中的我的信息这一栏，这一栏没有其他交互，纯信息展示。

![](data:application/octet-stream;base64,)

5-14 用户信息管理-我的信息

**描述：**

如上图5-14所示，用户在登录完成后，会进入首页，首页就是我的信息栏，其左侧(web后端管理登录后的所有页面皆是如此)是导航栏，分别有两个模块，导游信息管理和旅程管理。将鼠标上浮到对应的模块，模块将显示对应的选择，导游信息管理将显示我的信息和修改信息两栏。行程管理将显示我的行程，增加行程，行程统计和行程修改四个选择。

右边部分是各个选择对应的内容，首页就是我的信息对应的内容，这一块，可以看到该导游的各种信息，包括头像，姓名，id，邮箱，公司，职位，格言等信息。

**5.2.2.2 修改信息**

用户在登录后，将鼠标上浮左侧导航栏中的导游信息管理，将出现导游信息管理的选择，点击修改信息选项，进入修改信息的页面。

![](data:application/octet-stream;base64,)

5-15用户信息管理-修改信息

**描述：**

如上图5-15所示，此为用户信息修改界面，可以信息输入框分为两种，一种是浅灰的，一种是深灰色的，深灰色的输入框无法进行修改。 也就是只能修改姓名，邮箱和格言。 另外，点击上传头像，可以修改用户头像

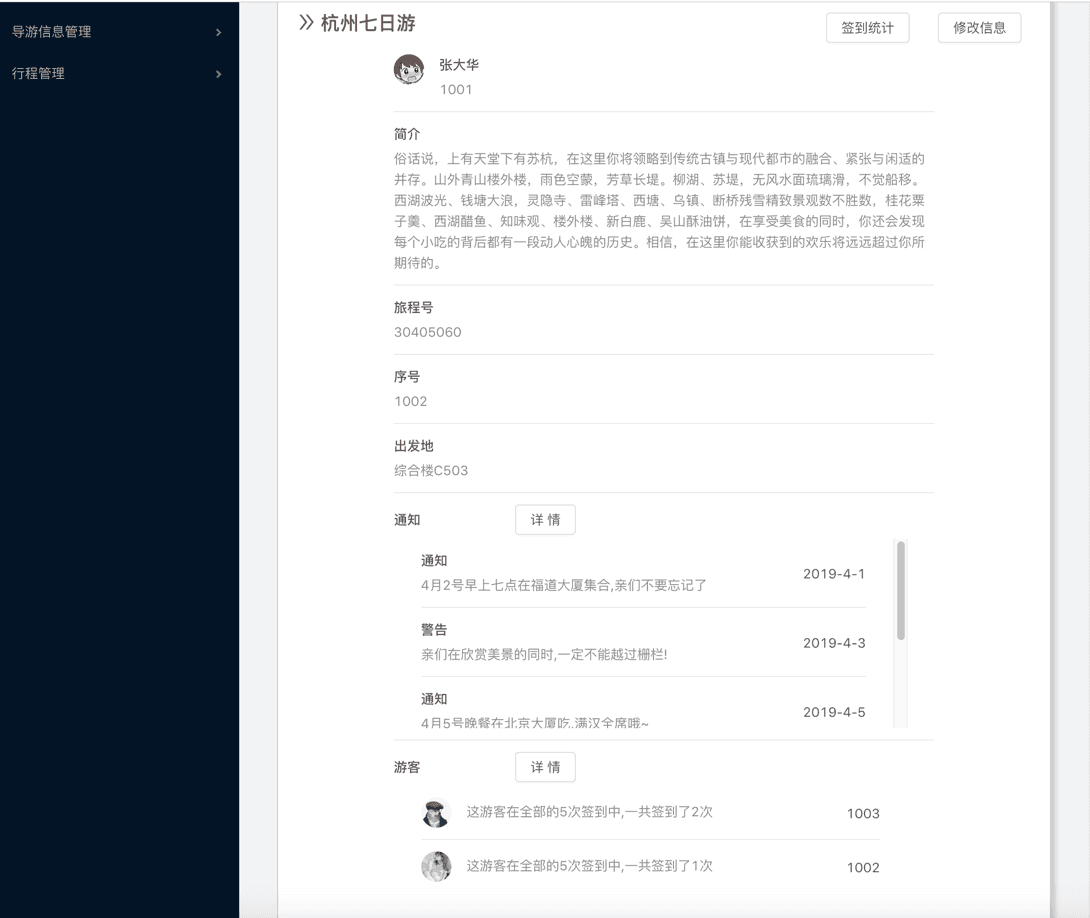
### 5.2.3 行程管理

**5.2.3.1 我的行程**

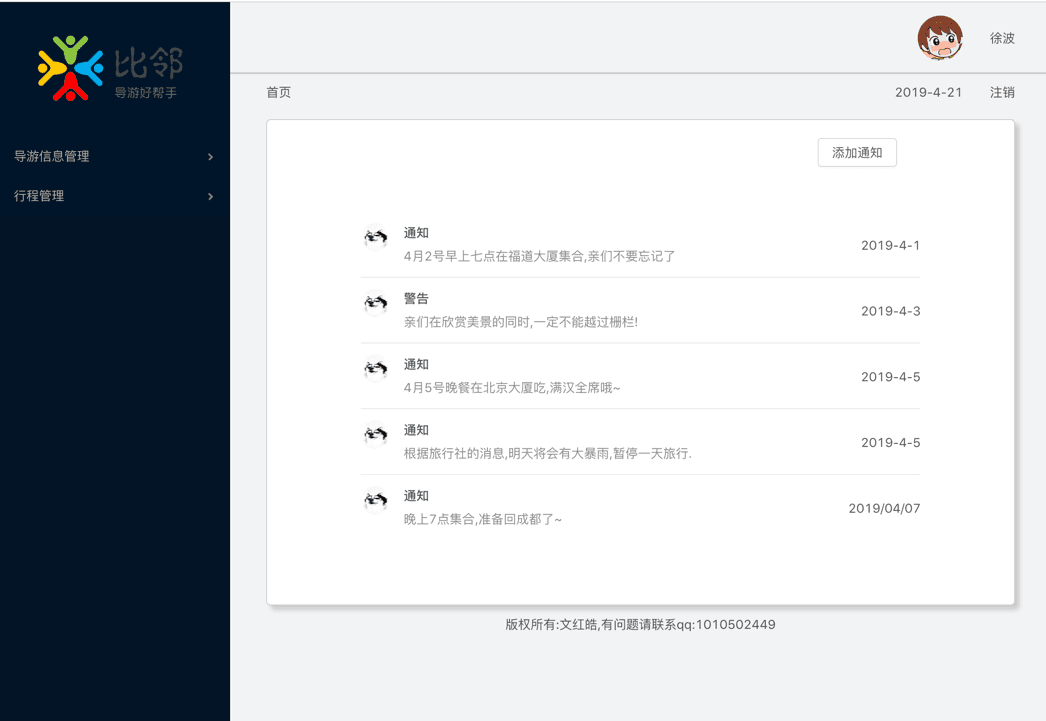
用户在登录后，将鼠标上浮左侧导航栏中的行程管理 ，将出现行程管理的选择, 点击我的行程选项，进入行程信息的页面。通过我的行程可以进入对应的行程详情页面，并进行相应的操作，比如添加删除通知和游客，修改行程信息等。

![](data:application/octet-stream;base64,)

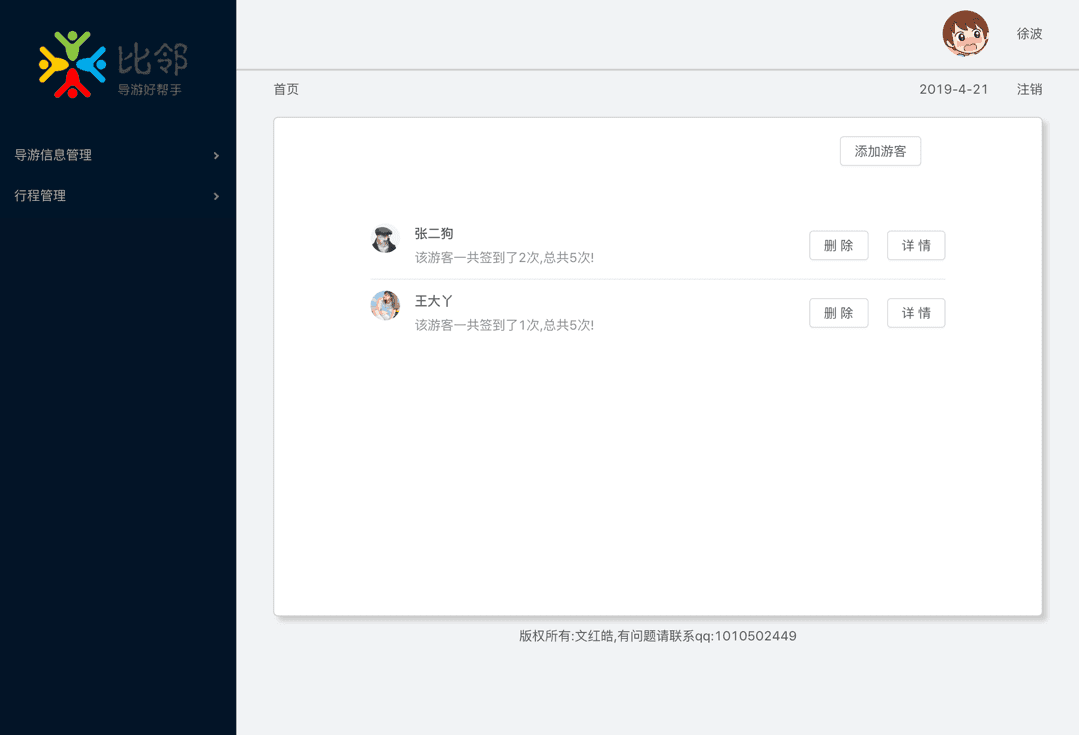
5-16 行程管理-我的信息



5-17 我的行程-行程详情



5-18 行程管理-通知管理



5-19 游客管理

**描述：**

如上图5-16所示，这是行程管理页面，这里简单的描述了此次行程的名称，简介, 旅程号，和旅程序号. 每个行程右下方有一个详情按钮，点击进入行程详情。

如图5-17所示，这是行程详情页面，这里详细的描述了所有和这次行程相关的所有信息，包括导游名，导游id，旅程号，序号，出发地，通知和游客。信息的右上角可以看到有签到统计和修改信息两个按钮，点击它们分别进入对应页面, 这里不详细阐述，5.2.3.3和5.2.3.4再详细阐述。通知和游客的信息处，分别有一个详情按钮。点进去后可以看详情并对相应内容进行操作。

如图5-18所示，点击通知的详情按钮，可以进入通知管理，这里包含了这个行程进行时，对应导游所发布的所有通知，每个通知可也看到它的标题，发布时间和具体内容。右上角有一个添加通知按钮，点击它可以添加新的通知。

如图5-19所示，点击游客的详情按钮，可以进入游客管理，这里包含了这个行程所有的游客信息，每个游客信息里包括一个删除和一个详情按钮，点击删除，可以删除游客，点击详情可以查看或者修改游客信息。右上角有一个添加游客的按钮，点击后，可以添加游客。

**5.2.3.2 添加行程**

用户在登录后，将鼠标上浮左侧导航栏中的行程管理，将出现行程管理的选择，点击添加行程选项，进入添加信息的页面。在这里输入必要的信息，可以创建一个新的旅程。

![](data:application/octet-stream;base64,)

5-20 添加行程

**描述：**

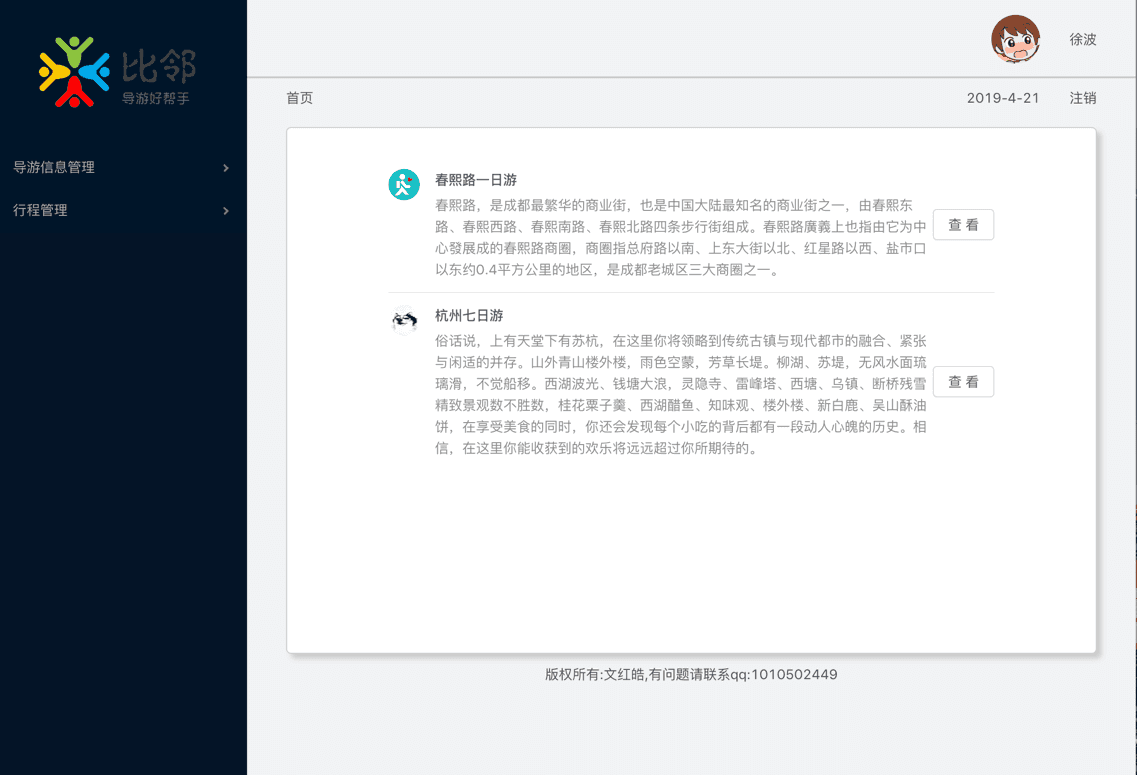
如上图5-20所示，此为添加行程的内容 . 输入必要的信息(旅程名，旅程号，序号，出发点，旅程首图). 然后点击确定，如果旅程号和序号没有同时与数据库已有旅程重复，系统将自动创建一个新的旅程。 创建好后,在我的行程哪里可以看到。

**5.2.3.3 行程统计**

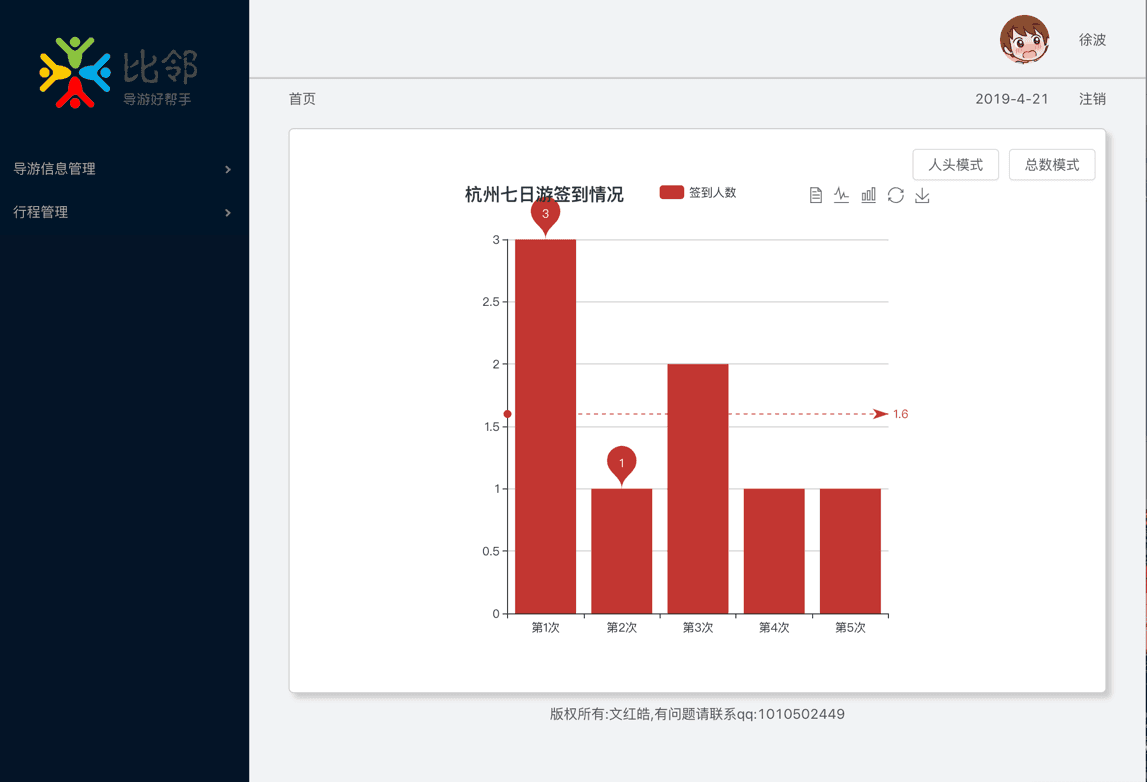
行程统计的入口有两个:

1.用户在登录后，将鼠标上浮左侧导航栏中的行程管理，将出现行程管理的选择，点击行程统计选项，进入行程统计首页。在首页选择对应的行程，点击查看按钮 ，进入详情页，就可以看到该行程所有的统计信息。

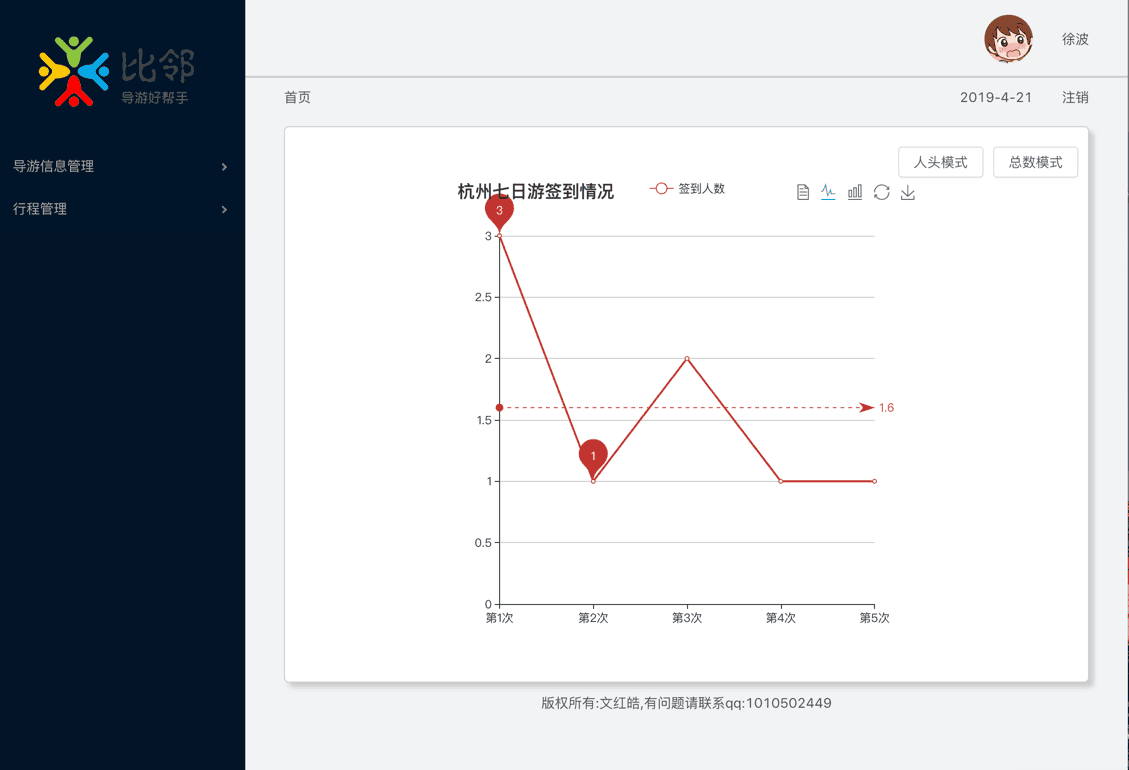
2.如5.2.3.1中所示，在行程详情里，点击签到统计按钮，也能够看到该行程的详细统计信息。



5-21 行程统计首页



5-22 行程统计详情-柱状图



5-23 行程统计详情-折线图

**描述：**

如图5-21所示，此为行程统计首页，点击对应的行程的查看按钮，进入统计详情页。

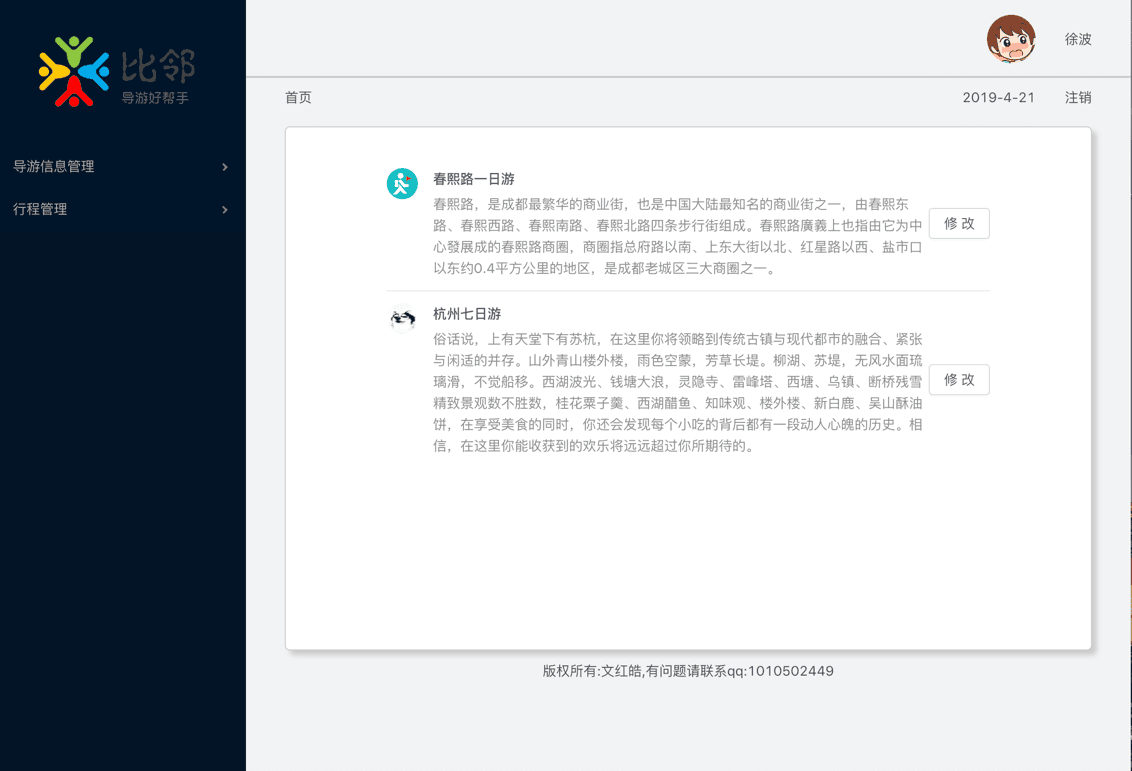
如图5-22和5-23 所示 ，这是行程统计详情页，通过可视化的方式展现了游客们的签到情况，可以选择柱状图和折线图等方式查看情况。

**5.2.3.4 行程修改**

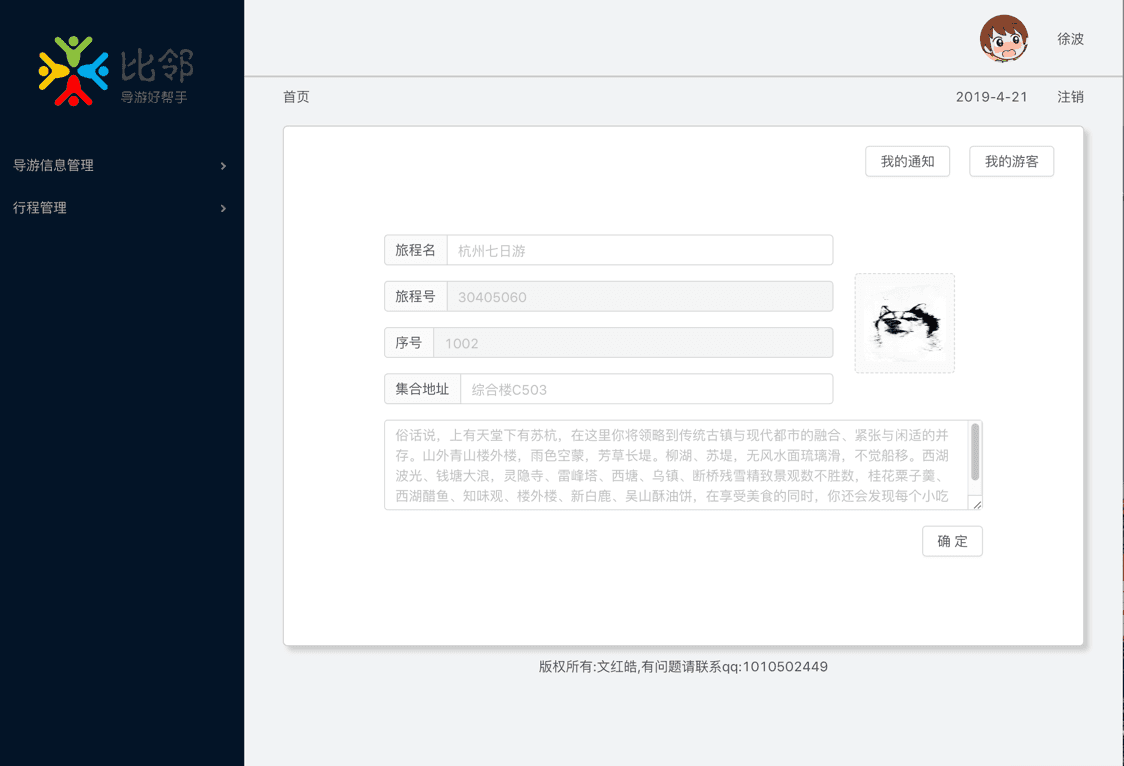
行程统计的入口有两个:

1.用户在登录后，将鼠标上浮左侧导航栏中的行程管理，将出现行程修改的选择，点击行程修改选项，进入行程修改首页。在首页选择对应的行程，点击修改按钮 ，进入详情页，就可以修改这个行程的信息了。

2.如5.2.3.1中所示，在行程详情里，点击修改信息按钮按钮，也能够看到该行程的详细信息，在这个页面也能进行修改。



5-24 行程修改首页



5-25 行程修改详情页

**描述：**

如图5-24所示，此为行程修改首页，点击对应的行程的修改按钮，进入修改详情页。

如图5-25所示，这是行程修改详情页，可以看到这里的输入框分为两种，浅色和深色的，浅色的是可修改想信息，深色的是不可修改的信息。填入对应的信息后，点击确定按钮即可完成修改。

## 5.3小结

本章主要是对系统的详细设计和实现。介绍了系统各的各个模块和功能的详细设计与实现情况，对于比较复杂的功能并附上了核心代码。用图示和文字的方式向读者展示了这个系统的实现结果和实现过程。

# 6 系统测试

系统完成后，就需要进行系统测试，系统测试是对整个系统做硬件，软件测试，检测它与预期是否一致。这里的系统测试只需要检测功能，测试功能是否符合预期结果。对于不符合预期的结果，需要后期修改以符合预期。

## 6.1 测试需求

测试需求见表6-1。

表6-1 测试需求表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **主模块** | **子模块** | **用例编号** | **用例名称** | **优先级** |
| 微信小程序 | 用户账号管理 | IITS-SRS-1.1 | 初次登录，用户账号绑定 | 1 |
| IITS-SRS-1.2 | 非初次登录，微信直接登录 | 1 |
| 签到管理 | IITS-SRS-2.1 | 导游打开签到 | 1 |
| IITS-SRS-2.2 | 导游关闭签到 | 1 |
| IITS-SRS-2.3 | 导游代签 | 1 |
| IITS-SRS-2.4 | 导游取消签到 | 2 |
| IITS-SRS-2.5 | 游客wifi签到 | 1 |
| 旅行日志 | IITS-SRS-3.1 | 查看旅游日志详情 | 2 |
| 通知管理 | IITS-SRS-4.1 | 导游发布通知 | 2 |
| IITS-SRS-4.2 | 用户查看通知 | 1 |

续表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Web  后端  管理  系统 | 注册登录 | IITS-SRS-5.1 | 注册 | 1 |
| IITS-SRS-5.2 | 密码账号登录 | 1 |
| IITS-SRS-5.3 | 手机验证码登录 | 1 |
| 用户信息管理 | IITS-SRS-6.1 | 用户信息查看 | 1 |
| IITS-SRS-6.2 | 用户信息修改 | 1 |
| 旅程管理 | IITS-SRS-7.1 | 旅程信息修改 | 1 |
| IITS-SRS-7.2 | 旅程信息查看 | 1 |
| IITS-SRS-7.3 | 添加通知 | 1 |
| IITS-SRS-7.4 | 增加旅程 | 1 |
| IITS-SRS-7.5 | 增加游客 | 1 |
| IITS-SRS-7.6 | 删除游客 | 1 |
| IITS-SRS-7.7 | 修改游客信息 | 1 |
| IITS-SRS-7.8 | 游客签到情况统计(图形化) | 1 |
| IITS-SRS-7.9 | 游客详情查看 | 1 |

## 6.2测试目的

系统测试的目的是按照需求文档检测所有功能是否符合预期，如果与预期结果相同，则不修改；如果与预期结果不同，需要修改以使项目符合预期。

## 6.3测试方法

本系统测试使用黑盒测试。黑盒测试只需要关注系统的输入和输出，按照系统的需求文档传入对应的合理的输入，观察理论输出是否符合期望，如果符合，则不需要做任何处理，如果不符合，则需要修改项目使项目所有输出符合期望。

## 6.4测试范围

测试范围主要报矿两部分，系统UI和系统功能。

系统UI主要是系统图标，拟态框，交互按钮等前端；系统功能是需求文档中所列出的所有功能。

## 6.5测试环境

测试环境见表6-2。

表6-2系统测试环境表

|  |  |  |
| --- | --- | --- |
| **硬件环境** | **微信小程序** | **Web后端管理系统** |
| 硬件配置 | 内存2g以上  带有wifi功能 | 处理器：Intel 酷睿M系列4代 CPU或者以上，主频2.10GHz以上  内存2g以上  带有wifi功能 |
| 软件配置 | Android5.0以上  微信版本5.3以上 | Windows7家庭版以上  Ie8+，chrome 70以上或者同等的近年已支持react的现代浏览器 |

## 6.6 测试用例

### 6.6.1微信小程序

微信小程序测试用例见表6-3。

表6.6.1微信测试用例及结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **测试功能** | **测试编号** | **操作** | **期望结果** | **实际结果** | **是否通过** |
| 用户账号绑定 | IITS-SRS-1.1.1 | 初次登录，输入正确的用户名和密码绑定账户 | 绑定成功 | 同期望 | 通过 |
| IITS-SRS-1.1.2 | 初次登录，输入错误的用户名和密码绑定账户 | 绑定失败 | 同期望 | 通过 |
| IITS-SRS-1.2 | 非初次登录，基于微信账户，直接登录 | 登录成功 | 同期望 | 通过 |
| 签到管理 | IITS-SRS-2.1.1 | 导游打开热点，随后点击开始签到按钮，输入一个周围存在的wifi | 找到正确的wifi并将此wifi信息存储到数据库 | 同期望 | 通过 |
| IITS-SRS-2.1.2 | 导游打开热点，随后点击开始签到按钮，输入一个周围不存在的wifi | 提示导游不存在此wifi | 同期望 | 通过 |
| IITS-SRS-2.2 | 点击关闭签到按钮 | 导游和游客都无法进行签到 | 同期望 | 通过 |
| IITS-SRS-2.3 | 导游在开始签到的情形下，点击游客头像旁边的签到按钮，进行代签 | 游客签到成功 | 同期望 | 通过 |

续表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | IITS-SRS-2.4 | 导游在开始签到的情形下，点击游客头像旁边的取消签到按钮，取消游客签到状态 | 游客签到状态被取消 | 同期望 | 通过 |
| IITS-SRS-2.5.1 | 游客点击签到，周围存在导游上传的wifi信息 | 签到成功 | 同期望 | 通过 |
| IITS-SRS-2.5.2 | 游客点击签到，周围不存在导游上传的wifi信息 | 签到失败 | 同期望 | 通过 |
| 旅行日志 | IITS-SRS-3.1 | 点击日志标题 | 显示日志具体内容 | 同期望 | 通过 |
| 通知管理 | IITS-SRS-4.1 | 导游发布通知 | 数据库能正确收到通知 | 同期望 | 通过 |
| IITS-SRS-4.2 | 用户查看通知 | 用户能看到通知的具体内容，包括发布时间，标题和内容 | 同期望 | 通过 |

### 6.6.2 web后端管理系统

Web后端管理系统测试用例见表6-4。

表6-4 web后端管理系统测试用例及结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **测试功能** | **测试编号** | **操作** | **期望结果** | **实际结果** | **是否通过** |
| 注册登录 | IITS-SRS-5.1.1 | 注册用户并输入完整信息，用户id与已有账户不冲突 | 注册成功 | 同期望 | 通过 |
| IITS-SRS-5.1.2 | 注册用户并输入完整信息，用户id与已有账户冲突 | 注册失败 | 同期望 | 通过 |
| IITS-SRS-5.1.3 | 注册用户并输入不完整信息 | 注册失败 | 同期望 | 通过 |
| IITS-SRS-5.2.1 | 密码账户登录，输入错误的密码和账户 | 登录失败 | 同期望 | 通过 |
| IITS-SRS-5.2.2 | 密码账户登录，输入正确的密码和账户 | 登录成功 | 同期望 | 通过 |
| IITS-SRS-5.3.1 | 手机号登录，输入错误的验证码 | 登录失败 | 同期望 | 通过 |
| IITS-SRS-5.3.2 | 手机号登录，输入正确的验证码 | 登录成功 | 同期望 | 通过 |
| 用户信息管理 | IITS-SRS-6.1 | 在导航栏进入我的信息 | 看到用户信息 | 同期望 | 通过 |
| IITS-SRS-6.2 | 进入修改信息页面，修改用户信息 | 修改成功 | 同期望 | 通过 |

续表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 旅程管理 | IITS-SRS-7.1 | 进入旅程信息修改页面，修改信息 | 修改成功 | 同期望 | 通过 |
| IITS-SRS-7.2 | 进入我的旅程页面 | 能看到我的所有旅程信息 | 同期望 | 通过 |
| IITS-SRS-7.3 | 在详情页里添加通知 | 添加通知成功 | 同期望 | 通过 |
| IITS-SRS-7.4.1 | 增加旅程，输入与已有旅程冲突的旅程号和序号 | 添加失败 | 同期望 | 通过 |
| IITS-SRS-7.4.2 | 增加旅程，输入与已有旅程不冲突的旅程号和序号 | 添加成功 | 同期望 | 通过 |
| IITS-SRS-7.5.1 | 增加游客，添加已在当前旅行团的游客 | 添加失败 | 同期望 | 通过 |
| IITS-SRS-7.5.2 | 增加游客，添加不在当前旅行团的游客 | 添加成功 | 同期望 | 通过 |
| IITS-SRS-7.6 | 删除游客 | 删除成功 | 同期望 | 通过 |
| IITS-SRS-7.7 | 修改游客信息 | 修改成功 | 同期望 | 通过 |
| IITS-SRS-7.8 | 查看该旅程签到情况，有柱状图，折线图两种方式 | 所有可视化方式数据均显示正常 | 同期望 | 通过 |
| IITS-SRS-7.9 | 查看游客详情 | 数据正常显示 | 同期望 | 通过 |

## **6.7** 测试结论

本次系统测试执行了系统用例35个，完全通过35个，0个未通过，测试成功率为100%。测试覆盖系统需求文档中的所有功能，包括微信小程序和web后端管理两端的模块，涵盖所有的用户，签到，通知和旅行日志交互操作。

通过对测试结果分析，可以得出该项目进度较好，基本完成项目需求文档中的所有功能，但是UI还需要美化。

## 6.8 小结

本章描述了系统UI测试和功能测试的流程，6.1简要介绍了测试需求，6.2-6.5描述了测试方法，目的，测试范围和测试环境，随后在6.6给出了具体的测试用例和每个用例的结果，最后在6.7给出了测试最终得出的结论，进展较为顺利。

# 总结与展望

该论文详细描述了我的毕设项目-基于wifi探针和node爬虫的导游助手的实现过程。

基于wifi探针做人员统计，而不是传统的点到在签到时间和效率上有较大的提高，减少了签到前的预先准备过程。通过mac地址和ssid的匹配，确保了人员统计过程的精确性，另外，与业界常用的wifi连接再进行签到统计相比，签到端可以同一时间段进行签到的用户数量更多。通知的发布查阅功能的引入也凸显了小程序的工具属性，使得导游在消息通知方面更加灵活。此外，通过node爬取马蜂窝，携程的游记信息并依据地理位置推送给游客，拓宽了小程序的交互场景，不仅只是单纯的工具，更是一个内容聚合平台。

为了方便导游的使用，在小程序之外，引入了web管理系统，一方面帮助导游管理游客和旅程的相关信息，另一方面，echarts的引入使得导游能通过图形化的方式看到游客人员统计与签到情况，更加的高效方便。

但在系统的测试过程中也遇到了一些问题，最明显的问题是小程序端ui设计的不统一，没有一致的风格，距离商用还有一定的距离。此外，在室内情况下，由于室内环境复杂，桌椅遮挡等的原因，在wif信号传递过程中，会受到墙壁，家具，门窗甚至人体等遮挡，wifi签到范围往往会很小，给签到过程带来了一定的困难。因此，还需要对系统进行不断的优化和改进。

# 参考文献

[1]中国报告网. 2018年中国智慧旅游行业分析报告—市场深度分析与投资前景研究[N]. 2018。

[2]高见,袁德嵛. 基于WIFI探针的预警系统设计与研究[J].中国人民公安大学, 2016.

[3]雷磊. 微信小程序开发入门与实践[M]. 北京：清华大学出版社, 2019.

[4]Kyle Simpson. 你不知道的JavaScript[M]. 人民邮电出版社, 2015.

[5]Alex Banks,Eve Porcello.React. 学习手册[M]. 中国电力出版社, 2017.

[6]Banker K. MongoDB in action[M]. Manning Publications Co. 2011.

[7]Ethan Brown.Node与Express开发[M]. 人民邮电出版社, 2017.

[8] John Arundel. Puppet. 实战指南[M]. 人民邮电出版社, 2017.

[9]百度. Echarts官方文档[db/ol], 2019.

[10]夏青. 大唐移动公司ERP项目实施方法评估与改进[D]. 北京邮电大学, 2012.

[11] 凌仲熙. 基于Android系统的共享停车位APP的设计与实现[D]. 四川大学, 2018.

[12] 陈光强. 基于Android的高校应届生求职App的设计与实现[D]. 2018.

# 附录

## 附录1：项目源代码说明

篇幅所限，不贴出。

具体代码请见github地址：<https://github.com/hughlovecode/guide>。

**致 谢**

经历了这几个月的毕业论文的和毕业设计的工作，我终于完成了这一篇毕业论文。这段时间压力很大，一边在实习，一边准备大创结题，一边还要准备我的毕设。百度实习的事情很多，虽然只是实习生，但是由于项目马上上线了，所以加了很多班，甚至在一定程度上影响了我的毕设。感激黄戈老师没有因为这个原因责怪我，反而尽心尽力的帮助我，不论是在前期的论文准备，论文选题，还是在后来的项目实现以及论文编写，我都是在老师的指导下完成的。

所以，我首先需要感谢的人是我的导师黄戈，感谢他给我的指导与帮助，也感谢他允许能一边实习，一边准备毕设. 没有他的帮助，我可能不能如此顺利的完成我的毕设。

当然，我还要感谢我的父母，我的同学们。感谢你们在生活上给予我的帮助，感谢你们在无数个黑夜，愿意花费时间听我倾泻压力。毕业论文的完成给我的大学生活划上了一个圆满的句号，但是大学对我的影响却不会到此为止，川大像一盏明灯，照耀在我的未来之路上。时间倏忽而过，转眼已经四年过去，我即将脱离校园这个象牙塔.愿我能乘风破万里浪，一往无前，登上凌绝顶。

**声** **明**

本人声明所呈交本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得四川大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

本学位论文成果是本人在四川大学读书期间在导师指导下取得的，论文成果归四川大学所有，特此声明。

学位论文作者（签名）

论文指导教师（签名）

年 月 日