

**本 科 毕 业 论 文（设 计）**

题目（中文） 基于微信小程序的旅游相册的设计与实现

（英文） The Design and Implementation of Travel Album Based on

Wechat Mini Program

学 院 信息与机电工程学院

年级专业 2017级计算机科学与技术

学生姓名

学 号

指导教师

**完 成 日 期 2020 年 4 月**

**上海师范大学本科毕业论文（设计）**

**诚信声明**

本人郑重声明：所呈交的毕业论文（设计），题目《基于微信小程序的旅游相册的设计与实现》是本人在指导教师的指导下，进行研究工作所取得的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明。除此之外，本论文（设计）不包含任何其他个人或集体已经发表或撰写过的作品成果。本人完全意识到本声明应承担的法律责任。

作者签名：

日期：2020年 4 月 日

**上海师范大学毕业论文（设计）指导记录表**

**学院：** 信息与机电工程学院

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 专业 | 计算机科学与技术（师范） | | 指导教师 |  | 学生姓名 |  | | 学号 |  |
| 毕业论文（设计）题目 | |  | | | | | | | |
| 日期 | 指导内容 | | | 存在问题与进一步改进意见 | | | 教师签名 | | 学生签名 |
|  |  | | |  | | |  | |  |
|  |  | | |  | | |  | |  |
|  |  | | |  | | |  | |  |
|  |  | | |  | | |  | |  |
|  |  | | |  | | |  | |  |
|  |  | | |  | | |  | |  |

注：本表由指导教师根据毕业论文（设计）指导工作方案和实际指导情况填写，在指导工作完成后交学院存档，保存期四年

**上海师范大学本科毕业论文（设计）选题登记表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 |  |  |  | 专业 | 计算机科学与技术 |
| 学 院 | 信息与机电工程学院 | 指导教师姓名**/**职称 | |  | |
| 题 目 | 基于微信小程序的旅游相册的设计与实现 | | | | |
| 本选题的意义及国内外发展简况：  随着物质水平的提高，越来越多的人们热衷于旅行，而在旅行时用手机拍照已经成为一种习惯，不断产生一波又一波新的照片。在手机内存有限的情况下，许多用户都会选择将自己照片存放在云相册或者网盘上，来节省手机内存的同时，方便管理备份自己的照片。但是，身为云相册类产品，除了满足用户备份或储存照片这种基础需求以外，还能为用户做些什么呢？怎么才能让自己变得更有价值？ 考虑到微信的广泛使用，如果采用微信小程序的方式，用户无需下载APP，使用体验较好。同类的相册类应用并不少见，但是功能相对单一，普遍不具备分享功能。基于以上分析，设计并实现基于微信小程序的旅行相册，具备以下功能：基础的存储和备份功能，此外，具有快速查找，智能分类，共享相册等功能。相册类产品还希望能够增加用户的使用时长和粘度。因此，产品要做到让图片浏览更沉浸，使相册成为用户回忆的一种方式。 | | | | | |
| 研究内容：  1. 进行合理的菜单划分和功能处理，开发出一个界面友好、使用方便、稳定可靠的微信小程序。  2. 使用HTML、CSS、JavaScript及微信原生组件等实现前端用户交互界面。  3. 使用MySQL数据库、Java及其开发框架SpringBoot来实现后端服务器。  4. 通过微信小程序开发平台、阿里云服务器、微信云开发、IntelliJ IDEA、Tomcat、Xshell 6、Navicat for MySQL、Postman等平台或工具完成系统开发。 | | | | | |
| 研究方法、手段及步骤：  1. 学习参考文献和网上资料与视频，研究并掌握软件开发时所要用到的若干关键技术。  2．按照软件工程思想，完成软件的需求分析、系统设计、详细设计与编码。  3. 对软件基本功能进行测试、优化和完善。  4. 撰写论文初稿，与指导老师沟通，不断修改论文结构与论述，最终定稿。 | | | | | |
| 主要参考文献：  [1] 科夫勒. MySQL 5 权威指南[M]. 北京: 人民邮电出版社, 2006:9~12  [2] 应用程序接口.百度百科[EB/OL]. http://baike.baidu.com/view/592964.htm  [3] 微信小程序开发者文档[EB/OL]. http://mp.weixin.qq.com/wiki  [4]葛萌,黄素萍,欧阳宏基.基于Spring MVC框架的Java Web应用[J].计算机与现代化,2018(08):97-101.  [5]刘红卫.微信小程序应用探析[J].无线互联科技,2016(23):11-12+40. | | | | | |
| 时间进度：  2020年11月： 布置毕业论文和说明要求、学生选题。  2020年11月20日： 完成选题和填写“选题登记表”。  2020年11月20日---12月30日 搜索资料，阅读，系统分析与设计。  2020年12月---2021年2月 系统实现，提交论文大纲，完成论文初稿  2021年3月底前 系统测试，完成论文修改稿。  2021年4月底前 进一步完善系统，论文最终定稿。  学生 （签名） 2020 年 11 月 18 日 | | | | | |
| 指导教师意见：  指导教师 （签名） 2020 年 月 日 | | | | | |
| 专业负责人意见：  专业负责人 （签名） 年 月 日 | | | | | |

注：本表与毕业论文（设计）一起存档，保存期为四年。

**摘 要**

随着智能手机的普及，人们习惯于用手机拍摄照片，但是越来越多的照片和日益庞大的手机App不断占用手机存储空间，本着减少手机存储空间占用的目标，设计一款云相册，来保存日益增长的用户照片，而触手可及，无需下载的微信小程序无疑时最好的选择。

本文设计了一款基于微信小程序的云相册，主要具备以下功能：云端个人相册的建立，相册图片的上传与下载，共享相册的分享，智能识别图片内容并进行自动分类。

云相册采用前后端的思想进行开发，系统主要由小程序客户端、服务器、存储服务器组成。前端使用微信开发者工具，小程序原生组件及API进行开发，后端使用Java语言，开发工具使用IDEA，使用Spring Boot框架，持久层框架使用MyBatis，数据库采用MySQL，并使用Maven来进行项目构建和管理，图片存储使用微信云开发提供的存储功能。

本文按照系统的设计与实现过程中，需求分析、数据库设计、系统设计与实现、系统测试的顺利来进行阐述。

关键词：微信小程序；图像识别；云相册；Spring Boot

**Abstract**

With the popularity of smart phones, people are used to taking photos with their mobile phones, but more and more photos and increasingly large mobile phone apps continue to occupy mobile phone storage space. In order to reduce the occupation of mobile phone storage space, a cloud photo album App is designed to save users' photos, Wechat Mini Program which have the advantages of within reach and does not need to be downloaded, so it is undoubtedly the best choice.

Design a cloud photo album App based on Wechat Mini Program, which mainly has the following functions: the establishment of cloud personal photo album, the upload and download of photo album pictures, the sharing photo album, intelligent recognition of image content and automatic classification.

Cloud photo album is designed with the idea of front and back end. The system is mainly composed of small program client, server and storage server. The front end uses Wechat developer tools, Wechat API for development, the back end uses Java language, the development tool uses IDEA, uses Spring Boot framework, the persistence layer framework uses MyBatis, the database uses MySQL, and uses Maven for project construction and management, and the image storage uses the storage function provided by Wechat cloud development.

In this paper, according to the system design and implementation process, demand analysis, database design, system design and implementation, system testing to elaborate.

**Keywords**: Wechat Mini Program；Cloud Album；Image Recognition；

Spring Boot

**目 录**

上海师范大学本科毕业论文（设计）诚信声明 ………………………………… I

上海师范大学本科毕业论文（设计）指导记录表…………………………………II

上海师范大学本科毕业论文（设计）选题登记表…………………………………III

中文摘要及关键词 …………………………………………………………………Ⅴ

英文摘要及关键词………………………………………………………………… Ⅵ

[第1章 绪论 1](#_Toc66120166)

[1.1 研究的背景 1](#_Toc66120167)

[1.2 研究的目的和意义 1](#_Toc66120168)

[1.2.1 研究目的 1](#_Toc66120169)

[1.2.2 研究意义 2](#_Toc66120170)

[1.3国内外技术现状分析 2](#_Toc66120171)

[1.4主要工作 4](#_Toc66120172)

[1.4.1 前端 4](#_Toc66120173)

[1.4.2 后端 4](#_Toc66120174)

[1.4.3 云服务部署 4](#_Toc66120175)

[第2章 相关概念与技术介绍 5](#_Toc66120176)

[2.1 相关概念 5](#_Toc66120177)

[2.1.1 微信小程序 5](#_Toc66120178)

[2.1.2 阿里云服务器 5](#_Toc66120179)

[2.1.3 IDEA 5](#_Toc66120180)

[2.1.4 Spring Boot 6](#_Toc66120181)

[2.1.5 MySQL 6](#_Toc66120182)

[2.2 相关技术及工具 7](#_Toc66120183)

[2.2.1前后端分离技术 7](#_Toc66120184)

[2.2.2 Postman 7](#_Toc66120185)

[2.2.3 Xshell 8](#_Toc66120186)

[2.2.4 Git 8](#_Toc66120187)

[2.2.5 Navicat for MySQL 8](#_Toc66120188)

[第3章 系统分析与设计 10](#_Toc66120189)

[3.1 可行性分析 10](#_Toc66120190)

[3.1.1经济可行性 10](#_Toc66120191)

[3.1.2技术可行性 10](#_Toc66120192)

[3.2 需求分析 11](#_Toc66120193)

[3.1.1 功能需求 11](#_Toc66120194)

[3.1.2 非功能性需求 12](#_Toc66120195)

[3.1.3 性能需求 13](#_Toc66120196)

[3.3功能分析 13](#_Toc66120197)

[3.3.1总体功能 13](#_Toc66120198)

[3.3.2用户模块 14](#_Toc66120199)

[3.3.3相册模块 14](#_Toc66120200)

[3.3.4图片模块 15](#_Toc66120201)

[3.3.5搜索模块 16](#_Toc66120202)

[3.4软件整体结构框架 16](#_Toc66120203)

[3.5 数据库设计 17](#_Toc66120204)

[3.5.1 E-R图 17](#_Toc66120205)

[3.5.2 用户信息表 18](#_Toc66120206)

[3.5.3 相册信息表 19](#_Toc66120207)

[3.5.4 图片信息表 19](#_Toc66120208)

[第4章 系统实现 21](#_Toc66120209)

[4.1 开发环境配置 21](#_Toc66120210)

[4.1.1微信小程序注册及开发工具安装 21](#_Toc66120211)

[4.1.2云服务器购买及配置 21](#_Toc66120212)

[4.1.3百度图像识别接口申请 22](#_Toc66120213)

[4.1.3软件版本 22](#_Toc66120214)

[4.2小程序开发 22](#_Toc66120215)

[4.2.1.页面设计 22](#_Toc66120216)

[4.2.1登录及权限获取 25](#_Toc66120217)

[4.2.2图片上传 26](#_Toc66120218)

[4.2.3搜索功能 27](#_Toc66120219)

[4.2.3相册及图片浏览 28](#_Toc66120220)

[4.2.4查看智能分类 30](#_Toc66120221)

[4.3后端开发 31](#_Toc66120222)

[4.4项目部署 33](#_Toc66120223)

[第5章 系统测试 34](#_Toc66120224)

[5.1测试环境 34](#_Toc66120225)

[5.2测试用例 35](#_Toc66120226)

[5.2.1用户登录 35](#_Toc66120227)

[5.2.2个人相册浏览 35](#_Toc66120228)

[5.2.3共享相册浏览 36](#_Toc66120229)

[5.2.4新建相册 36](#_Toc66120230)

[5.2.5相册设置 37](#_Toc66120231)

[5.2.6上传图片 37](#_Toc66120232)

[5.2.7下载图片 38](#_Toc66120233)

[5.2.8图片信息修改 38](#_Toc66120234)

[5.2.9查看分类结果 39](#_Toc66120235)

[5.3性能测试 40](#_Toc66120236)

[第6章 总结与展望 41](#_Toc66120237)

[6.1总结 41](#_Toc66120238)

[6.2展望 41](#_Toc66120239)

[参考文献 43](#_Toc66120240)

[附录 程序代码 45](#_Toc66120241)

[附录一.小程序 45](#_Toc66120242)

[主要函数： 45](#_Toc66120243)

[页面展示设计： 50](#_Toc66120244)

[附录二.后端代码 51](#_Toc66120245)

[GetController 51](#_Toc66120246)

[GetService： 53](#_Toc66120247)

[GetMapper 53](#_Toc66120248)

# 第1章 绪论

## 1.1 研究的背景

随着物质水平的提高和信息技术日新月异的发展，当今世界变得越来越便利，智能手机已经是现代人生活的不可分割的一部份。人们在满足基本的物质需求的基础上，越来越热衷于精神需求的满足，旅游成为了人们的不二之选。手机摄影技术的飞速发展，也使得以往门槛高且昂贵的摄影变得越来越大众化，不再需要专业的单反，人们用口袋里的手机就可以完成拍摄，成像效果足以媲美一些专业相机。

随着旅行时用手机拍照已经成为一种习惯，不断产生一波又一波新的照片。而现在的智能手机往往无法扩展存储空间，在手机内存有限的情况下，许多用户都会选择将自己照片存放在云相册或者网盘上，来节省手机内存的同时，方便管理备份自己的照片。同时，人们热衷于使用微信朋友圈来分享照片[1]，因此手机里往往需要这么一款云相册类应用来满足用户存储并分享照片的应用。

## 1.2 研究的目的和意义

### 1.2.1 研究目的

身为云相册类产品，除了满足用户备份或储存照片这种基础需求以外，还能为用户做些什么呢？怎么才能让自己变得更有价值？考虑到微信在中国的广泛使用，基本上人人都有微信，因此，如果采用微信小程序的方式，用户无需下载专用APP，更加节省手机内存空间，得益于微信提供的接口，用户也可以得到很好的使用体验。

考虑到同类的相册类应用并不少见，但是功能相对单一，普遍不具备分享功能。根据以上分析，设计基于微信小程序的旅行相册，具备以下功能，首先是基础的存储和备份功能。相册类产品还希望能够增加用户的使用时长和粘度。因此，产品要做到让图片浏览更沉浸，使相册成为用户回忆的一种方式，所以还具有分享功能，可以发布自己的旅游照片，与天南海北的网友分享。

该产品还应具有智能识别功能，根据用户上传的图片，AI智能识别图像内容并进行分类。后台数据存储到云服务器，使得数据安全得到保障，同时降低运维成本。

### 1.2.2 研究意义

采用小程序的形式，用户无需专门下载App，而且依托于微信平台，可以便捷的分享该应用。相册保存到云端，避免因手机故障等原因造成数据遗失。

使用图像识别技术，对用户上传的照片进行分类，用户可根据关键词进行搜索，更快速的在海量图片中找到自己需要的图片。将后台服务器及数据库部署到阿里云上，降低后期维护成本，同时，用户照片使用微信云开发体提供的存储功能进行管理，保证用户加载图片的速度，也可通过云服务控制台进行可视化管理。

## 1.3国内外技术现状分析

随着APP人口红利进入尾声，用户的增长迈入了瓶颈期。退潮之时，最先做出应对的互联网巨头们，以自身拥有的超级APP为基础搭建小程序或轻应用的分发平台。据腾讯最新财报显示，微信月活跃用户突破 12 亿，微信小程序更加受到众多开发者和用户的喜爱。小程序以小见大，能满足用户简单的需求，而且小程序将会与APP本体互相作用，互利共生，一起进步[2]-[4]。

市面上有许多相册类应用[5]-[8]，不过大多为App形式，需要用户下载软件才可以使用，难免有些不方便，云相册实现的原理大同小异，微信小程序提供的云开发功能比较方便[9]，因此如果将云相册和微信小程序相结合，能带来更好的效果。

2006年8月9日，Google首席执行官埃里克·施密特（Eric Schmidt）在搜索引擎大会（SES San Jose 2006）首次提出“云计算”（Cloud Computing）的概念，在此之后云服务蓬勃发展，成为众多互联网巨头布局的重点内容之一，亚马逊云（AWS）和微软云（Azure）占据了国外公有云的主要市场分额，国内各大头部互联网企业也纷纷发力，布局云服务，阿里云，腾讯云等得到了广泛使用。

近年来，随着人工智能的兴起，得益于神经网络，图像识别技术得到飞速发展，厂家也是将云服务和人工智能相结合，以百度为例，推出了AI开放平台，提供图像识别、人脸识别、身份证识别、文字识别等一系列端到端的软硬一体的应用,可以通过调用提供的接口，将图像识别能力与系统相结合。

Java是由Sun公司于1995年5月推出的Java程序设计语言和Java平台的总称。Java平台由Java虚拟机(Java Virtual Machine)和Java 应用编程接口(Application Programming Interface、简称API)构成。面向对象技术被称为程序设计思想的一场革命，它已成为计算机应用开发领域的主流趋势，Java语言是面向对象技术成功应用的范例之一。Java语言的面向对象、简单性、安全性、跨平台等显著特点，使得Java成为许多应用系统的理想开发语言。基于Java，许多公司制作了一些框架和工具便于开发，JetBrains公司推出的集成开发环境IDEA ( IntelliJ IDEA )在业界得到广泛好评，是最流行的Java 开发工具之一。IDEA的代码自动提示、重构、JavaEE支持、各类版本工具、JUnit、CVS整合、代码分析极大的便利了程序开发，提高了开发效率。

Linux是一个多用户、多任务、支持多线程和多CPU的操作系统，由于它免费使用和自由传播的优点，伴随着互联网的发展，Linux得到了来自全世界软件爱好者、组织、公司的支持。开发者往往选择Linux的发行版来作为服务器的系统。

Spring框架是一个开放源代码的J2EE应用程序框架，Spring解决了开发者在J2EE开发中遇到的许多常见的问题，提供了功能强大IOC、AOP及Web MVC等功能。MVC 模式代表 Model-View-Controller（模型-视图-控制器） 模式。这种模式用于应用程序的分层开发。小程序的具体开发也是基于MVC模式。Spring Boot基于Spring设计，继承了Spring框架原有的优秀特性，简化了Spring应用的整个搭建和开发过程。Spring Boot通过集成大量的框架解决依赖包的版本冲突和引用的不稳定性等问题[10]-[13]。

## 1.4主要工作

### 1.4.1 前端

使用微信官方提供的微信开发者工具进行前端页面的布局设计及功能实现包括页面跳转、获取用户信息、显示相册及图片、上传下载照片、相册设置、搜索图片等功能。其中，用户及相册信息调用后台服务器提供的接口获取，照片文件通过微信云开发提供的存储功能保存到云端[14]。

### 1.4.2 后端

使用Java集成开发环境IDEA进行后端代码编写，使用Sping Boot框架开发，用户及相册和图片信息使用Mysql数据库[20]进行保存，数据库使用Mybatis持久化框架，图片文件保存使用微信云存储。后台提供用户信息查询、相册及图片信息的查询及修改、关键词搜索等功能的接口[16]。

### 1.4.3 云服务部署

购买阿里云的轻量应用服务器进行部署，服务器系统采用CentOS 7.3，购买域名并将之解析到阿里云服务器的IP地址，通过访问域名来调用服务器功能。

# 第2章 相关概念与技术介绍

## 2.1 相关概念

### 2.1.1 微信小程序

微信小程序是一种不需要下载安装即可使用的应用，它实现了应用“触手可及”的梦想，用户扫一扫或者搜一下即可打开应用。也体现了“用完即走”的理念，用户不用关心是否安装太多应用的问题。应用将无处不在，随时可用，但又无需安装卸载。对于开发者而言，微信小程序开发门槛相对较低，难度不及APP，能够满足简单的基础应用，适合生活服务类线下商铺以及非刚需低频应用的转换。微信小程序能够实现消息通知、线下扫码、公众号关联等功能。其中，通过公众号关联，用户可以实现公众号与微信小程序之间相互跳转。小程序提供了一个简单、高效的应用开发框架和丰富的组件及API，帮助开发者在微信中开发具有原生 APP 体验的服务[17]-[21]。

### 2.1.2 阿里云服务器

云服务器Elastic Compute Service（ECS）是阿里云提供的一种基础云计算服务。使用云服务器ECS就像使用水、电、煤气等资源一样便捷、高效。无需提前采购硬件设备，而是根据业务需要，随时创建所需数量的云服务器ECS实例。在使用过程中，随着业务的扩展，可以随时扩容磁盘、增加带宽。如果不再需要云服务器，也能随时释放资源，节省费用。ECS涉及的所有资源，包括实例规格、块存储、镜像、快照、带宽和安全组。可以通过云服务器管理控制台或者阿里云 App 配置ECS资源。

### 2.1.3 IDEA

IDEA 全称 IntelliJ IDEA，是Java语言开发的集成环境，IntelliJ在业界被公认为最好的Java开发工具之一。Intellij IDEA 最突出的功能自然是调试（Debug)，可以对Java代码，JavaScript，JQuery，Ajax等技术进行调试。此外，Intellij IDEA具有以下优点：智能选取、丰富的导航模式、历史记录功能、对重构的优越支持和编码辅助，Java规范中提倡的toString()、hashCode()、equals()以及所有的get/set方法，你可以不用进行任何的输入就可以实现代码的自动生成，从而把你从无聊的基本方法编码中解放出来。

采用Intellij IDEA进行开发，它灵活的排版功能、XML的完美支持、动态语法检测、代码检查、对JSP的完全支持、智能编辑、完美的自动代码完成、版本控制完美支持、不使用代码的检查、智能代码、正则表达式的查找和替换功能**、**程序员意图支持，可以极大提高开发效率。

### 2.1.4 Spring Boot

Spring Boot是由Pivotal团队在2013年开始研发、2014年4月发布第一个版本的全新开源的轻量级框架。它基于Spring设计，不仅继承了Spring框架原有的优秀特性，而且还通过简化配置来进一步简化了Spring应用的整个搭建和开发过程。另外Spring Boot通过集成大量的框架使得依赖包的版本冲突，以及引用的不稳定性等问题得到了很好的解决。Spring Boot具备以下特征：

（1）可以创建独立的Spring应用程序，并且基于其Maven或Gradle插件，可以创建可执行的JARs和WARs；

（2）内嵌Tomcat或Jetty等Servlet容器；

（3）提供自动配置的“starter”项目对象模型（POMS）以简化Maven配置；

（4）尽可能自动配置Spring容器；

（5）提供准备好的特性，如指标、健康检查和外部化配置；

（6）绝对没有代码生成，不需要XML配置。

### 2.1.5 MySQL

MySQL 是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 公司。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL 支持大型的数据库。可以处理拥有上千万条记录的大型数据库。

MySQL 使用标准的 SQL 数据语言形式。

MySQL 可以运行于多个系统上，并且支持多种语言。这些编程语言包括 C、C++、Python、Java、Perl、PHP、Eiffel、Ruby 和 Tcl 等。

MySQL 对PHP有很好的支持，PHP 是目前最流行的 Web 开发语言。

MySQL 支持大型数据库，支持 5000 万条记录的数据仓库，32 位系统表文件最大可支持 4GB，64 位系统支持最大的表文件为8TB。

## 2.2 相关技术及工具

### 2.2.1前后端分离技术

随着互联网的高速发展以及IT开发技术的升级，前后端分离已成为互联网项目开发的业界标准使用方式。在实际工作中，前后端的接口联调对接工作量占HTML5大前端人员日常工作的30%-50%，甚至会更高。

什么是前后端分离？把前端与后端独立起来去开发，放在两个不同的服务器，需要独立部署。前端通过接口来调用后端的API，前端只需要关注页面的样式与动态数据的解析和渲染，而后端专注于具体业务逻辑。

前后端分离具有以下优点：彻底解放前端、提高工作效率，分工更加明确、局部性能提升、降低维护成本、实现高内聚低耦合，减少后端（应用）服务器的并发/负载压力、可以使后台能更好的追求高并发、高可用、高性能，使前端能更好的追求页面表现、速度流畅、兼容性、用户体验等。

前端工程调用接口的方式：Ajax在浏览器与Web服务器之间使用异步数据传输（HTTP 请求），通过get或者post等方法与服务器进行交互。

### 2.2.2 Postman

接口是客户端和服务器来进行交互的，接口返回的数据一般都是json格式的数据类型。在前后端分离开发时，后端工作人员完成系统接口开发后，需要与前端人员对接，测试调试接口，验证接口的正确性可用性。而这要求前端开发进度和后端进度保持基本一致，任何一方的进度跟不上，都无法及时完成功能模块的测试。作为后端开发人员，要求独立开发完成某个接口后，开发人员自己需要先测试通过后再提交给测试人员进行测试，否则会出现到测试人员哪里业务流程根本就走不通，或者BUG会过多的情况等。Postman就是这样一款接口测试工具，能够高效的帮助后端开发人员独立进行接口测试。

### 2.2.3 Xshell

Xshell是一个强大的安全终端模拟软件，它支持SSH1, SSH2, 以及Microsoft Windows 平台的TELNET 协议。Xshell 通过互联网到远程主机的安全连接以及它创新性的设计和特色帮助用户在复杂的网络环境中享受他们的工作。Xshell可以在Windows界面下用来访问远端不同系统下的服务器，从而比较好的达到远程控制终端的目的。除此之外，其还有丰富的外观配色方案以及样式选择。通过使用Xshell,可以配置并管理云服务器。

### 2.2.4 Git

Git是目前世界上最先进的分布式版本控制系统。Git 是一个开源的分布式版本控制软件,用以有效、高速的处理从很小到非常大的项目版本管理。Git 最初是由Linus Torvalds设计开发的，用于管理Linux内核开发。GitHub是一个基于Git的远程文件托管平台。Git本身完全可以做到版本控制，但其所有内容以及版本记录只能保存在本机，如果想要将文件内容以及版本记录同时保存在远程，则需要结合其他软件来使用。

通过使用GitHub来管理项目，可以从服务器上克隆完整的Git仓库（包括代码和版本信息）到本地，在自己的机器上根据不同的开发目的，创建分支，修改代码。使用Git可以很好的进行版本控制和协同开发。

### 2.2.5 Navicat for MySQL

Navicat for MySQL是一套管理和开发MySQL的理想解决方案，支持单一程序，可同时连接到多个MySQL数据库。这个功能齐备的前端软件为数据库管理、开发和维护提供了直观而强大的图形界面，给MySQL新手以及专业人士提供了一组全面的工具。

# 第3章 系统分析与设计

## 3.1 可行性分析

可行性分析是开发一个软件的前提条件，对软件开发具有重要意义。因为在软件的开发过程中，会遇到各种难以预测的问题，包括技术上和经济上或者其他一些方面的问题，给项目开发带来很大的阻碍，往往会导致项目延期，甚至项目中止，造成人力物力的很大浪费，在软件开发前进行可行性分析，预测估计开发过程中可能遇到的问题，对潜在的技术难点制定解决方案，如果经过分析发现项目实现代价过大或者难以实现，在项目开始前就中止项目，避免项目开发到一半才终止所造成的巨大损失，以下就从经济可行性和技术可行性两个方面进行分析。

### 3.1.1经济可行性

经济可行性将开发系统所需要投入的资源和项目所产出的汇报进行粗略比较，理想情况下，系统产出大于投入，开发风险小，收益回报高。

本项目依托于微信小程序，使用微信原生组件进行开发，后端采用Spring Boot框架进行搭建，和传统App相比，微信小程序是跨平台的，不用针对IOS 和Andorid 分别开发两个版本，维护成本和开发成本都比传统App低。经过分析可得，该项目在经济上具有可行性。

### 3.1.2技术可行性

技术可行性指的是开发本系统所涉及到的技术是否可以得到满足，不超出目前的技术水平，完成系统开发，技术可行性分析是项目能否顺利开发的决定性因素之一。

本项目采用的微信小程序，虽然属于较新技术，但是微信官方提供了开发文档，内容全面，开发上没有太大的难点，小程序提供的API也大大提高了开发效率，后端使用的Spring Boot 本就是为了简化项目配置而产生的，内置了Tomcat服务器，本系统涉及到的项目代码编写比较容易，项目部署也不存在繁琐的步骤，因此，该项目在技术上也存在可行性。

## 3.2 需求分析

### 3.1.1 功能需求

对于一款云相册类APP，需要具备以下功能：相册浏览、相册新建及删除、相册信息修改、图片浏览、图片上传、图片下载，此外，本项目还具有搜索、智能识别、自动分类功能。

用户通过微信账号进行登录，通过微信用户唯一标识来进行认证，认证通过后，可以浏览共享相册和私人相册，共享相册只允许进行查看及下载，不能进行修改，新建个人相册后，可以进行图片上传、设置相册是否为共享、相册删除、相册封面设置、相册名修改等操作，对于个人相册中的图片，可以进行删除、图片标签设置和下载操作。

对于用户上传的每一张图片，在上传时，都会调用图像识别的API进行图像识别，与图片其他信息一同保存到数据库中，用户可通过搜索功能搜索图像识别结果及自定义标签来快速查找图片。

用户可以设置每一张图片的标签，然后系统会根据图片的位置信息进行自动分类，从而使得用户可以快速查看某一地点的图片。

用例图如图3-1所示

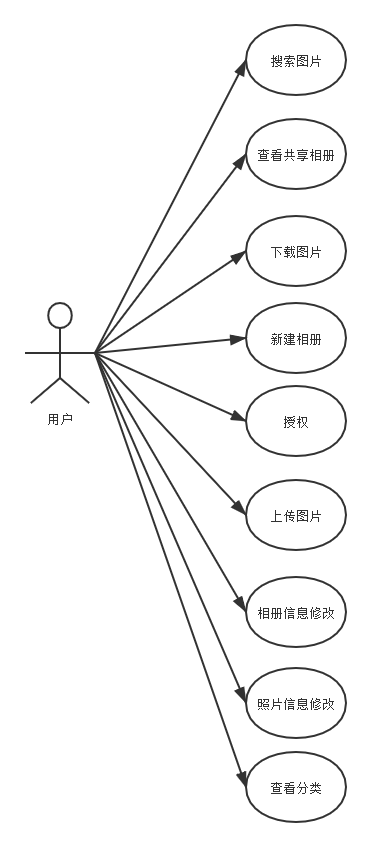


图3-1 用例图

### 3.1.2 非功能性需求

除了需要考虑到上图的功能性需求，主要需要考虑到以下几个方面的非功能性需求：

易用性需求，云相册运行在微信App上，通过微信提供的认证就可进行登录，免去了手动注册和登录过程，整个云相册的界面也十分简单明了，新用户都可以快速上手使用。

可靠性需求，项目使用的后台服务器使用阿里云服务器进行搭建，图片存储在微信云存储，避免了个人搭建服务器可能发生的服务器故障等意外所造成的数据丢失，此外，阿里云和微信云存储都可支持服务器扩容，当用户数量增加，遇到高并发请求时，可实时升级服务器，降低高负载所带来的风险。

安全性，小程序依托于微信App,对用户的个人信息需要进行授权认证，相册权限获取也需要用户确认，安全性上有保障。

此外，还需要考虑界面友好、可移植性等方面的非功能性需求。

### 3.1.3 性能需求

对于一个系统，响应速度是很重要的，作为相册类应用，需要对图片进行上传下载操作，更是对网络带宽有一定的要求，项目采用图片数据和图片存储分离的模式，服务器只负责数据库方面的操作，图片以路径的形式存储，图片实际存储微信云存储上，这样，图片的显示与获取进行了分离，即使网络状态不佳，图片的信息数据少，也可进行传输，而图片的显示则由微信进行异步刷新，最大程度的避免了因网络状况不佳所造成的卡顿。

## 3.3功能分析

### 3.3.1总体功能

项目主要可以分为四个模块，分别是用户登录、相册、图片、搜索模块，系统整体功能如图3-2所示。

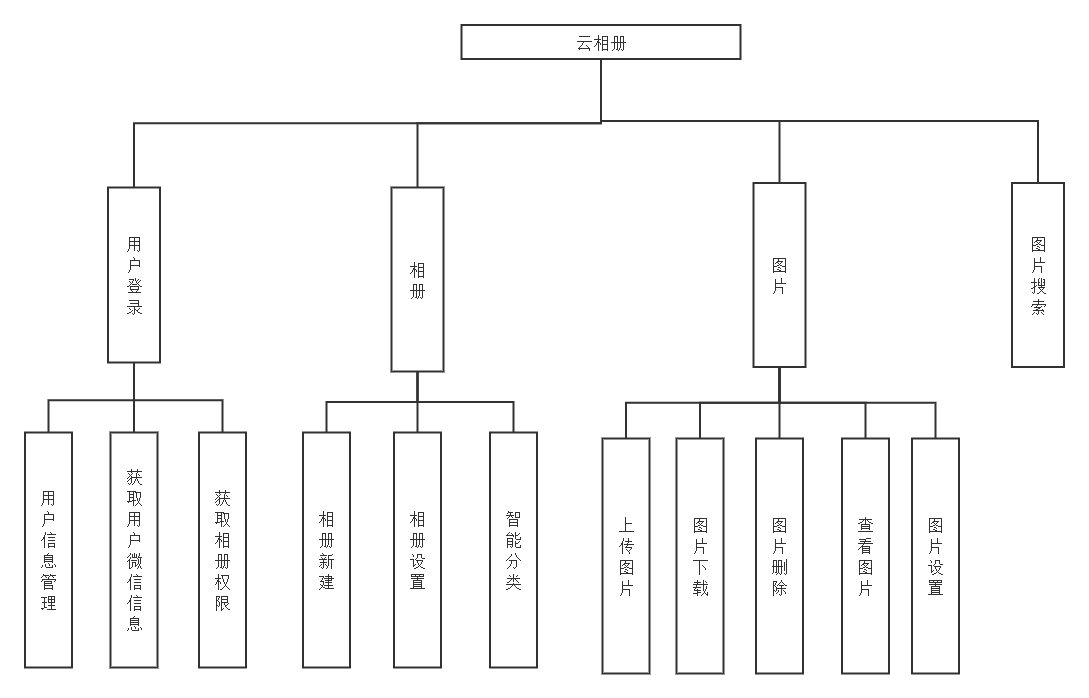


图3-2 系统功能图

### 3.3.2用户模块

用户模块主要包括登录认证、权限许可、用户信息修改。

1. 登录认证：当用户打开微信，进入旅游云相册小程序时，会自动获取用户身份凭证，如果该用户是新用户，则将该用户自动注册并添加到用户表中，如果是已注册用户，则展示用户个人相册。
2. 权限许可：当需要将图片保存到本机相册时，需要向用户获取相册权限，用户点击确定后才可成功保存，如果用户已经给过权限了，则会自动跳过该过程。
3. 用户信息修改：用户可对个人信息进行修改。

### 3.3.3相册模块

相册模块主要包括新建相册、相册信息修改、删除相册、上传相册图片

* 1. 新建相册：新建相册时，输入相册名和相册设置，增加新的个人相册
  2. 相册信息修改：当用户需要更新相册信息时，可对相册信息进行修改
  3. 删除相册：当用户不想要该相册时，想一次性删除该相册全部图片，则删除相册。
  4. 智能分类：小程序会根据图像识别出图片的结果以及用户自定义的标签，对用户的图片进行分类，将相同主题的图片归为同一类。

### 3.3.4图片模块

1. 上传图片：对于个人相册或者共享相册，能够进行多张图片的上传并浏览。上传图片时的流程如图3-3所示。

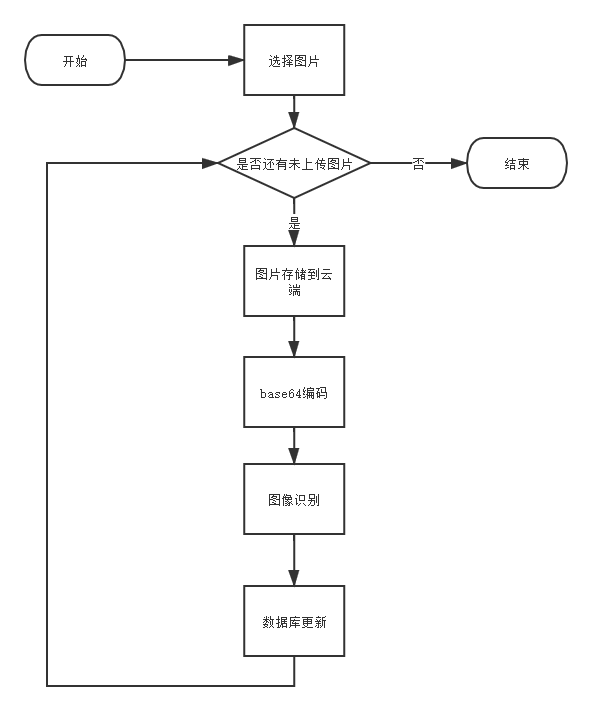


图3-3 图片上传流程图

1. 图片下载：用户可以将云端的个人相册或其他人的共享相册中的图片下载到本地
2. 图片删除：用户可删除个人相册中不想要的图片
3. 图片标签设置：用户可自定义每张图片的标签，如果图像识别结果有误，也可进行手动修改

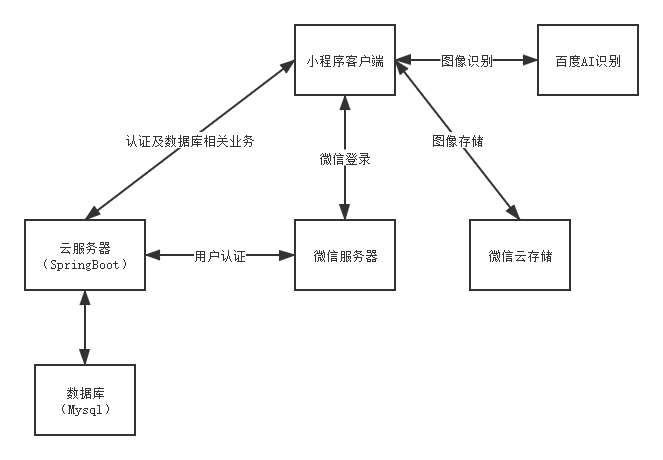
### 3.3.5搜索模块

搜索模块主要实现搜索功能，用户可输入关键词，根据用户给图片打上的标签或者图像识别的结果，返回匹配的图片，便于用户快速找到自己想要的图片。

## 3.4软件整体结构框架

本应用采用了小程序的形式，因此采用前后端分离进行开发，软件主体采用B/S架构，小程序负责视图层的渲染，从后端提供的接口获取数据并展示，后端服务器提供用户登录、相册及图片查询、搜索等接口。后端返回的数据统一为JSON格式，此外还需用到微信云开发提供的存储功能，百度AI的图像识别功能，数据库采用MySQL,该数据库简单易用，安全小巧，开发灵活便利。

当用户打开小程序的时候，小程序会获取用户的code,向微信服务器进行认证并获取微信用户唯一标识openID,再通过后端服务器，从数据库中查询相关数据，返回给小程序，小程序根据返回的数据，进行页面视图的渲染。上传图片时，客户端先调用相应接口，将图片保存微信云存储并进行编码，小程序再调用百度图像识别的接口进行图像识别，得到图像识别数据后，通过wx.request向后端服务器传输数据，进行数据库相应的更新。项目部署如图3-4所示。

图3-4 部署方案图

## 3.5 数据库设计

### 3.5.1 E-R图

数据库设计是软件开发的重要内容，关于本小程序的开发，涉及到的表有三张，分别为用户表、相册表、图像表，实体关系如图3-5所示。

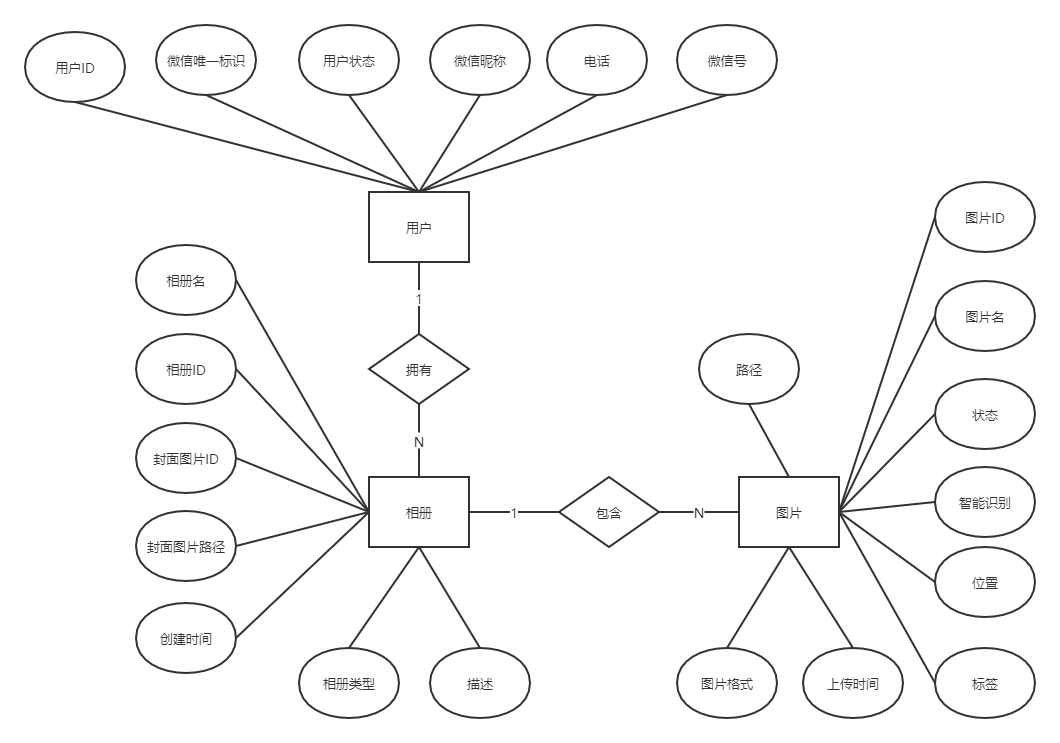


图3-5 E-R图

### 3.5.2 用户信息表

用户信息表，主要用于用户登陆时验证，其中用户id、微信唯一标识、用户状态用来身份认证，用户昵称、电话号码、微信号则用来保存用户信息。用户状态默认为normal，如果用户恶意上传非法图片，可以通过将ststus改为illegal禁止该用户登录。用户id为主键，并且自增。用户信息表如下所示。

表3-1 用户信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 字段名称 | 主键 | 字段含义 | 字段类型 | 是否可空 | 默认值 | 是否唯一 |
| 1 | userid | 是 | 用户id | INT(16) | 否 | 无 | 是 |
| 2 | openeid | 否 | 微信用户唯一标识 | VARCHAR(30) | 否 | 无 | 是 |
| 3 | status | 否 | 用户状态 | VARCHAR(10) | 否 | normal | 否 |
| 4 | nickname | 否 | 用户昵称 | VARCHAR(30) | 是 | 无 | 否 |
| 5 | phonenumber | 否 | 电话号码 | VARCHAR(11) | 是 | 无 | 否 |
| 6 | wechatid | 否 | 微信号 | VARCHAR(30) | 是 | 无 | 否 |

### 3.5.3 相册信息表

相册信息表保存相册信息，相册id为主键，自增，ststus字段记录相册是否被删除，delete表示相册已被删除，normal表示相册未被删除，删除相册后，相册该字段改为delete，数据库中仍保留该相册信息，这样用户误操作后，还可进行手动恢复，相册类型字段记录是否为共享相册，share表示该相册为共享相册，normal表示该相册为个人相册，不允许用户进行自定义，用户设置相册为共享或取消共享时，会更新此字段，表结构如下所示。

表3-2 相册信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 字段名称 | 主键 | 字段含义 | 字段类型 | 是否可空 | 默认值 | 是否唯一 |
| 1 | albumid | 是 | 相册id | INT(11) | 否 | 无 | 是 |
| 2 | userid | 否 | 相册所属用户id | INT(11) | 否 | 无 | 否 |
| 3 | status | 否 | 是否删除 | VARCHAR(10) | 否 | normal | 否 |
| 4 | albumname | 否 | 相册名 | VARCHAR(30) | 是 | 无 | 否 |
| 5 | coverimageid | 否 | 封面图片id | INT(11) | 是 | 无 | 否 |
| 6 | coverimagepath | 否 | 封面图片路径 | VARCHAR(255) | 是 | 无 | 否 |
| 7 | albumtype | 否 | 相册类型 | VARCHAR(10) | 否 | 无 | 否 |
| 8 | description | 否 | 相册描述 | text | 是 | 无 | 否 |

### 3.5.4 图片信息表

图片信息表记录图片信息，其中path为微信云存储的实际保存路径，ai字段记录图像识别结果，labe为用户自定义标签，ststus字段记录图片是否被删除，其他信息如下所示。

表3-3 图片信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 字段名称 | 主键 | 字段含义 | 字段类型 | 是否可空 | 默认值 | 是否唯一 |
| 1 | imgeid | 是 | 照片id | INT(11) | 否 | 无 | 是 |
| 2 | albumid | 否 | 所属相册id | INT(11) | 否 | 无 | 否 |
| 3 | imagename | 否 | 图片名 | VARCHAR(30) | 否 | 无 | 否 |
| 4 | path | 否 | 图片存储路径 | VARCHAR(255) | 否 | 无 | 否 |
| 5 | location | 否 | 位置 | VARCHAR(30) | 是 | 无 | 否 |
| 6 | ai | 否 | 图像识别 | VARCHAR(30) | 是 | 无 | 否 |
| 7 | label | 否 | 标签 | VARCHAR(30) | 是 | 无 | 否 |
| 8 | type | 否 | 图片格式 | VARCHAR(10) | 否 | 无 | 否 |
| 9 | status | 否 | 图片状态 | VARCHAR(10) | 否 | normal | 否 |

# 第4章 系统实现

## 4.1 开发环境配置

### 4.1.1微信小程序注册及开发工具安装

开发微信小程序的第一步是申请注册一个小程序AppID，需要到微信公众平台通过邮箱进行申请，如果没有注册小程序AppID的话，可以先使用测试号，但是无法使用微信云开发功能。注册好AppID后，下载并安装微信开发者工具，这是微信官方提供的开发工具。除此之外，还需安装Java集成开发工具IntelliJ IDEA，这是业界最流行的Java开发工具之一，教育版向学生用户免费开放，只需要用教育邮箱申请即可使用。由于SpringBoot内置了tomcat，所以无需进行tomcat的安装配置。此外还需安装以下软件：

1. Mysql数据库
2. Xshell，便于远程控制操作云服务器
3. Postman，进行接口调试
4. Navicat 12 for MySQL，管理维护数据库
5. Git，进行版本控制

### 4.1.2云服务器购买及配置

购买阿里云服务器，我购买的是轻量应用服务器，足够项目使用，云服务器安装JDK 和MySQL数据库，由于微信小程序对安全性的要求，还需要为后端服务器配置域名，购买xinyu001.top域名，由于相关法律法规要求，域名还必须进行工信部备案，完成备案后，将购买的的域名xinyu001.top解析到云服务器的IP地址，并为云服务器购买免费的SSl证书，全部完成后，小程序可以访问<https://xinyuoo1.top>来使用后端服务器的接口。

需要在阿里云控制台设置安全组规则，开放80端口用于接口访问，开放3306端口进行数据库访问。

### 4.1.3百度图像识别接口申请

项目需要调用百度图像识别接口，在百度AI开放平台申请通用物体和场景识别高级版API，得到API Key和Secret Key，后续通过post请求进行调用。百度提供一天500次的免费调用，超出后需要进行付费。

### 4.1.3软件版本

项目使用到的软件及框架版本如表4-1所示

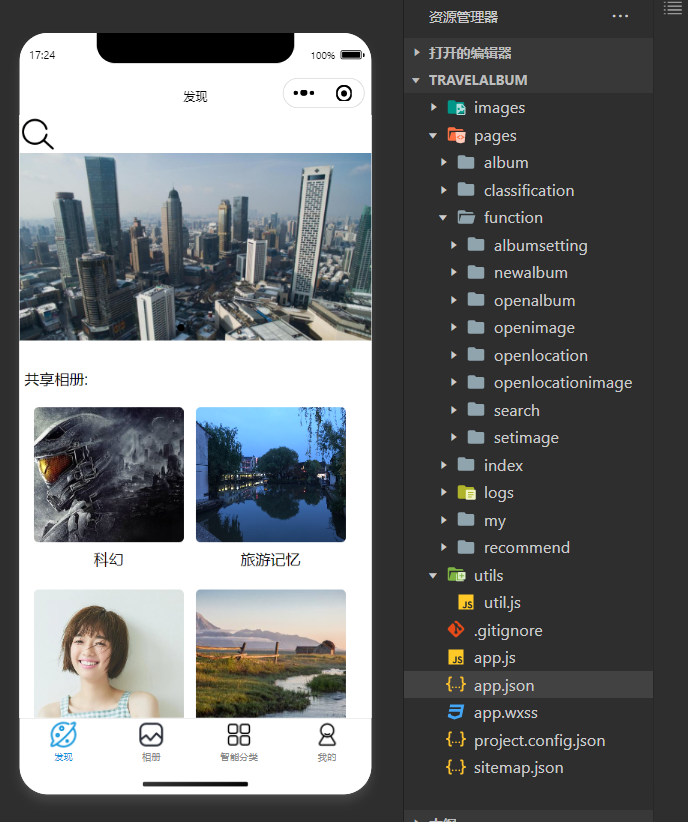
表4-1开发环境配置表

|  |  |
| --- | --- |
| 操作系统 | Windows 10 |
| 微信开发者工具 | 1.05.2102010 |
| IntelliJ IDEA | 2020.3.2 |
| MySQL | 5.7.33 |
| SpringBoot | 2.4.2 |
| JDK | 1.8.0\_281 |
| Xshell | 6.0.0039 |
| Postman | v8.0.6 |
| Apache Tomcat | 9.0.41 |
| 云服务操作系统 | CentOS 7.3 |

## 4.2小程序开发

### 4.2.1.页面设计

主页左上方为搜索按钮，点击后进入搜索跳转到搜索页面，页面底部为导航栏，分别是主页、个人相册、智能分类和我的四个页面，点击后可以进行跳转。主页上方为轮播图，可以滑动显示图片，下方则是共享相册，上下滑动查看。主页效果图和页面结构如图4-1所示。

图4-1 主页效果图及文件结构

每个小程序页面都需要使用到四个文件，后缀分别为js、json、wxss、wxml。其中js脚本文件负责逻辑层的实现，wxml和wxss文件控制页面的样式，实现视图层，json文件存储页面配置数据，每个页面最上方的标题就是在这个文件里配置。每个页面都有一系列默认函数， onLoad( ) 在第一次显示页面的时候调用，以后再次切换到该页面时则会调用onShow( )。

在app.js文件中配置小程序的全局变量，app.json中注册页面和底部导航栏的设置，定义在 app.wxss 中的样式为全局样式，作用于每一个页面。在 page 的 wxss 文件中定义的样式为局部样式，只作用在对应的页面，并会覆盖 app.wxss 中相同的选择器。使用微信云开发存储功能时，只需要输入图片对应路径地址，就可以像访问本地图片一样，直接访问，无需进行其他配置，十分方便。本项目用户上传的图片根据用户id分别建立文件夹进行存储，图片名使用上传时间加随机数和文件类型进行字符串相加来命名。小程序个人相册页面和智能分类页面如图4-2，图4-3所示。

图4-2个人相册 图4-3智能分类

### 4.2.1登录及权限获取

 小程序端首先调用 wx.login() 获取临时登录凭code，并回传到后端服务器。后端服务器调用 auth.code2Session 接口，换取 用户唯一标识 OpenID 和 会话密钥 session\_key，然后在进行后续一系列相关业务。流程图如图4-4所示。

图4-4小程序登录流程时序图[22]

小程序端定义三个函数，分别为getcode ( ) 、getopenid ( )、getuser( ) ，当小程序首次加载时，先调用getcode( ) 函数，使用wx.login( ) 获取code，在接口调用成功的回调函数中再调用getopenid( )函数，将code发送给后端服务器获取openid，成功后再调用getuser( )从后端获取用户信息，并将结果通过getApp()接口保存到全局数据中，便于后续使用。

 相册权限获取先通过 wx.getSetting（）获取用户当前的授权状态，用if语句进行判断，如果未授权相册权限，则调用wx.authorize( )申请相册权限。权限申请及获取用户信息如图4-5，图4-6所示。

图4-5获取用户信息 图4-6申请相册权限

### 4.2.2图片上传

首先使用wx.chooseImage()接口上传图片，一次最多上传9张，可以选择图片是否压缩和从本地相册选择图片或使用相机拍照。接口根据选择的图片数量返回一个临时路径数组tempFilePaths[ ]，数组大小与图片数量一致。将临时路径数据用this.setData( )进行保存到ImageTemPath中。

得到图片路径数组后，依次遍历数组，使用wx.getImageInfo获取图片类型，保存到数组type[ ]中，用Date().getTime()获取系统时间，Math.floor(Math.random() \* 1000)获取一个随机数，将图片类型与系统时间和随机数组合成图片名，将图片名保存到对应的数组imagename[ ]中。

生成图片名后，使用wx.cloud.uploadFile（）进行图片的上传，上传时需要指定目标路径和图片源路径，图片源路径从ImageTemPath中得到，使用app.globalData.user.userid从全局变量中获取到用户id从而确定目标文件夹，再加上imagename[ ]中的图片名，就是目标路径。

由于图像识别不支持jpeg或png格式的图片，支持的图片需要进行base64编码。上传图片到云端后，使用wx.getFileSystemManager().readFile()图片进行编码，将编码结果保存到base64[ ]数组中。

在调用图像识别接口前，还需要获取到一个access\_token，通过ApiKey 和SecretKey向<https://aip.baidubce.com/oauth/2.0/>发起POST请求得到access\_token，进行保存。得到access\_token后，就可以向<https://aip.baidubce.com/rest/2.0/image-classify/v2/advanced_general>调用图像识别接口，请求格式为POST,需要带上之前得到的base64格式的图片和access\_token。将接口返回的数据保存到ai[ ]数组。

完成以上一系列过程后，小程序将图片信息一并发送给后端服务器对应的接口，服务器将上传图片的信息保存到数据库中，从而完成上传图片的过程。上传图片如图4-7、图4-8所示。

图4-7上传图片 图4-8上传图片成功

### 4.2.3搜索功能

点击主页左上角的搜索图标，可以进入搜索页面，可以根据图片的位置信息，图像识别出的图像内容和用户添加的标签进行搜索，效果如图4-9、图4-10所示。

图4-9搜索结果1 图4-10搜索结果2

### 4.2.3相册及图片浏览

当小程序启动时，用户登录认证后得到用户信息，小程序使用wx.request向后端服务器发起get请求，把用户ID参数包含在URL中,后端会返回一个JSON格式的文件，里面保存了相册信息，小程序再根据相册信息，进行页面的渲染。

获取相册信息函数写在onshow中，每当重新切换或回退页面时都会进行相册数据的刷新，保证数据同步。

从主页共享相册和个人相册都可以进入相册页，从个人相册进入相册时，可以删除相册、相册设置、上传图片操作，从共享相册进入时，只能进行浏览和上传图片操作，小程序会根据用户打开的是个人相册还是共享相册，设置sharetag为不同值，再使用wx:if判断sharetag实现对删除相册、相册设置按钮的自动隐藏。

点击上传图片进行图片的上传。

点击删除按钮后，通过wx.showModal显示一个提示框询问是否确定删除，用户点击确定后，可以删除此相册，通过弹窗询问避免用户误操作。用户确定删除时，小程序向后端发起http请求，调用相关接口，传递相册id和用户id进行相册的删除。

点击相册设置按钮，使用wx.navigateTo跳转到查看图片页面，同时传递相册id和图片id。

打开共享相册如图4-11所示，打开个人相册如图4-12所示。

图4-11打开个人相册 图4-12打开共享相册

点击图片跳转到图片详情界面后，根据图片id向后端服务器发起http请求，获取图片详细信息，包括图片路径、图片地点、图片AI识别结果、用户自定义标签。用户可以将图片设置为该相册封面、删除图片、修改图片信息、下载图片。

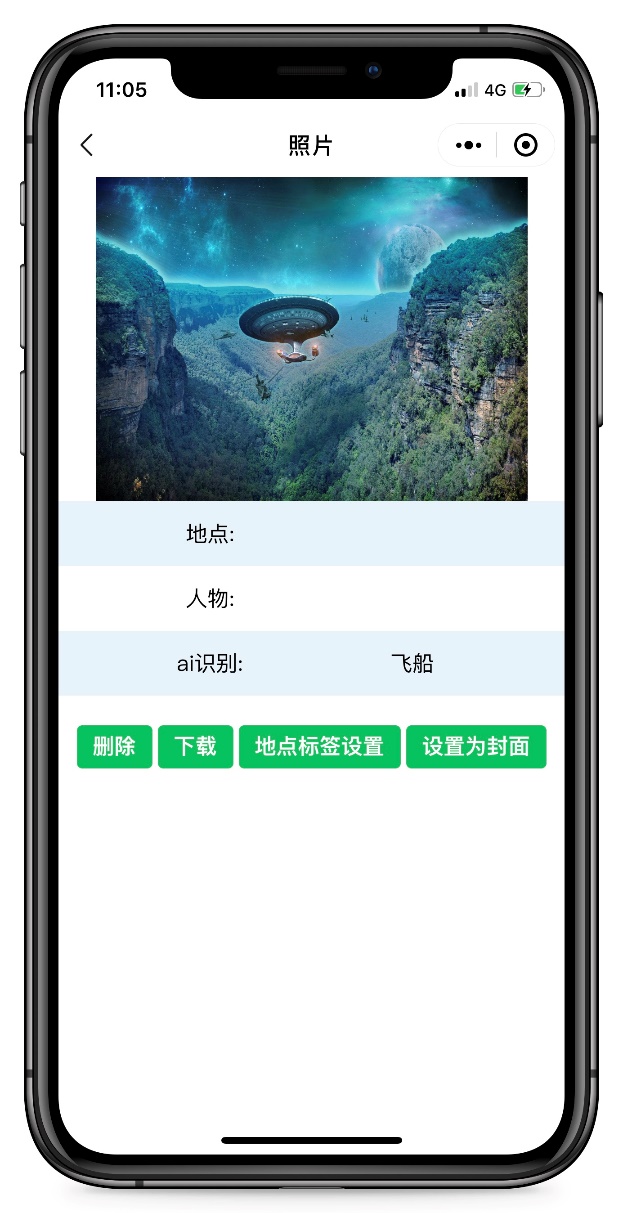
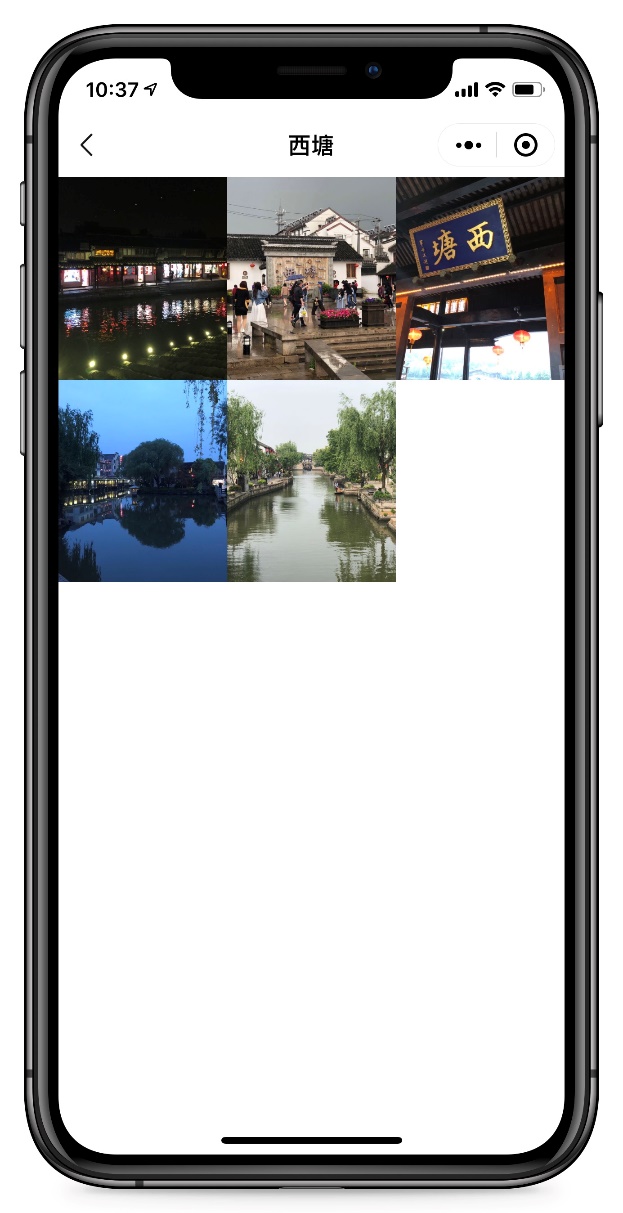
 从共享相册打开图片和从个人相册打开图片后跳转的图片详情界面会根据sharetag自动对相关按钮进行隐藏，保证用户相册安全。打开图片如图4-13、图4-14所示。

图4-13打开个人相册的图片 图4-14打开共享相册的图片

### 4.2.4查看智能分类

打开智能分类界面，会向后端服务器发起GET请求，获取相关数据，得到三个数组，分别是label[ ]、locationlist[ ]、ailist[ ]，小程序根据数组中的内容进行页面的渲染，如图4-15。以打开位置分类中的西塘为例，打开效果图如图4-16所示，点击具体图片，可以查看图片详情。

 图4-15智能分类主界面 图4-16查看分类

## 4.3后端开发

后端服务器主要为小程序提供接口，使用Spring Boot框架，用Maven来进行项目构建和管理。

TravelablumApplication是SpringBoot的启动类，在controller目录下新建GetController、SetController、UploadController来对请求进行转发分配，并新建对应的Service、Mapper类。建立Album、Image、User模型，通过操作模型来对数据库进行增删修改。项目开发使用到了以下注解：

@RestController，实现返回JSON格式的信息。

@RequestMapping()，实现对不同接口地址的路由转发。

@RequestParam，实现对参数的传递。

在项目配置文件application.yml中配置数据库和SSl证书的安装。

后端服务器实现的主要接口如表4-X所示

表4-1接口详情表

|  |  |  |
| --- | --- | --- |
| 接口URL地址 | 主要参数 | 实现作用 |
| /Get/OpenId | code | 获取用户openid |
| /Get/User | opened | 获取用户信息 |
| /Get/Albums | userid | 获取用户相册 |
| /Get/Album | albumid | 获取相册详细信息 |
| /Get/Images | albumid | 获取相册内图片 |
| /Get/Image | imageid | 获取图片信息 |
| /Get /Search | ai、label、location、userid | 根据关键词搜索图片 |
| /Get /Locations | userid | 获取地点分类结果 |
| /Get /ImagesByLocation | userid、location | 获取某一地点分类下包含的图片 |
| /Upload/Image | albumid、imagename、path、ai | 上传图片 |
| /Upload/Album | userid、albumname | 新建相册 |
| /Set/DeleteImage | imageid | 删除图片 |
| /Set/DeleteAlbum | albumid | 删除相册 |
| /Set/UpdateUser | userid、name… | 修改用户信息 |
| /Set/UpdateAlbum | albumtype、albumid、albumname、description | 修改相册信息 |
| /Set/UpdateImage | imageid、ai、location、label | 修改图片信息 |
| /Set/CoverImage | imageid、albumid | 设置图片为相册封面 |

## 4.4项目部署

代码编写完成后，小程序端的代码直接在微信开发者工具中上传为体验版，管理员可添加微信用户作为体验成员来体验小程序，小程序的正式版需要经过审核。

后端服务器的代码打包成jar格式，使用Xshell连接到云服务器，使用rz命令上传文件，使用nohup java -jar travelablum-0.0.1-SNAPSHOT.jar > /dev/null 2> /dev/null & 来启动服务器程序并保持运行。

# 第5章 系统测试

在系统开发完成后，最后一步就是测试，这是必不可少的。通过系统测试，确保系统开发实际结果和开发目标一致，保证系统开发质量。

## 5.1测试环境

本应用为微信小程序，考虑到兼容性，需要分别测试安卓平台和IOS平台的实际效果，用到的测试机器和服务器信息如下所示：

安卓系统：

* 设备厂商：小米
* 设备型号：红米4
* Android版本：Android 8
* CPU：高通骁龙625
* 内存：3GB
* 微信版本：8.0.1

IOS系统：

* 设备厂商：苹果
* 设备型号：iPhone X
* IOS版本：IOS 13.7
* CPU：A11+A11协处理器
* 内存：3GB
* 微信版本：8.0.2

服务器：

* 云服务器厂商：阿里云
* 操作系统：CentOS 7.3
* 内存：2GB
* 公网带宽：5M

## 5.2测试用例

### 5.2.1用户登录

用户登录测试用例如表5-1所示。

表5-1 用户登录小程序测试用例

|  |  |
| --- | --- |
| 用例名称 | 用户登录小程序 |
| 目的 | 测试用户打开小程序时微信登录和授权功能 |
| 前提 | 用户首次打开小程序 |
| 测试步骤 | 1)打开小程序，进入“我的”页面  2)弹出微信授权获取信息界面，选择允许或拒绝 |
| 预期结果 | 用户点击允许时，显示用户微信头像及昵称；用户拒绝时，显示默认微信头像和昵称 |
| 实际结果 | 实际结果同预期结果一致 |

### 5.2.2个人相册浏览

个人相册浏览测试用例如表5-2所示。

表5-2 个人相册浏览测试用例

|  |  |
| --- | --- |
| 用例名称 | 个人相册 |
| 目的 | 测试个人相册浏览功能 |
| 前提 | 用户登录小程序 |
| 测试步骤 | 1)进入相册页面  2)点击相册，进入相册详情页 |
| 预期结果 | 跳转到相册详情页，显示该相册内图片 |
| 实际结果 | 实际结果同预期结果一致 |

### 5.2.3共享相册浏览

共享相册浏览测试用例如表5-3所示。

表5-3 共享相册浏览测试用例

|  |  |
| --- | --- |
| 用例名称 | 共享相册 |
| 目的 | 测试共享相册浏览 |
| 前提 | 无 |
| 测试步骤 | 1)点击主页的共享相册 |
| 预期结果 | 跳转到共享相册详情页面，显示共享相册内图片 |
| 实际结果 | 实际结果同预期结果一致 |

### 5.2.4新建相册

新建相册测试用例如表5-4所示。

表5-4 新建相册测试用例

|  |  |
| --- | --- |
| 用例名称 | 新建相册 |
| 目的 | 测试新建相册功能 |
| 前提 | 用户登录小程序 |
| 测试步骤 | 1)打开相册页面，点击新建相册  2)跳转到新建相册页面，输入相册信息 |
| 预期结果 | 输入相册名并选择相册类型后，点击确定，新建相册成功，系统自动返回相册页面 |
| 实际结果 | 实际结果同预期结果一致 |

### 5.2.5相册设置

相册设置测试用例如表5-5所示。

表5-5 相册设置测试用例

|  |  |
| --- | --- |
| 用例名称 | 相册设置 |
| 目的 | 测试相册设置功能 |
| 前提 | 用户登录小程序 |
| 测试步骤 | 1)打开相册页面，选择个人相册进入详情页  2)点击删除相册按钮 |
| 预期结果 | 弹窗询问是否确定删除，再次点击确定后删除此相册，系统返回相册页面 |
| 实际结果 | 实际结果同预期结果一致 |

相册封面设置测试用例如表5-6所示。

表5-6 相册封面设置测试用例

|  |  |
| --- | --- |
| 用例名称 | 封面设置 |
| 目的 | 测试相册封面设置 |
| 前提 | 用户登录小程序，打开个人相册 |
| 测试步骤 | 1)打开相册页面，选择个人相册进入详情页  2)点击要设置为封面的图片 |
| 预期结果 | 点击设置为封面按钮，回到相册页面，相册封面设置成功 |
| 实际结果 | 实际结果同预期结果一致 |

### 5.2.6上传图片

上传图片测试用例如表5-7所示。

表5-7 上传图片测试用例

|  |  |
| --- | --- |
| 用例名称 | 图片上传 |
| 目的 | 测试图片上传 |
| 前提 | 打开相册 |
| 测试步骤 | 1)点击上传图片按钮  2)选择或拍摄图片 |
| 预期结果 | 图片上传成功后相册内自动刷新出新上传的图片 |
| 实际结果 | 实际结果同预期结果一致 |

### 5.2.7下载图片

下载图片测试用例如表5-8所示。

表5-8 下载图片测试用例

|  |  |
| --- | --- |
| 用例名称 | 下载图片 |
| 目的 | 测试下载图片 |
| 前提 | 打开个人相册或共享相册 |
| 测试步骤 | 1)点击图片  2)跳转到图片详情页，点击下载按钮 |
| 预期结果 | 弹窗获取用户相册权限，如果已经获取过权限，则会跳过权限获取，图片成功保存到本地 |
| 实际结果 | 实际结果同预期结果一致 |

### 5.2.8图片信息修改

图片信息修改测试用例如表5-9所示。

表5-9 图片信息修改测试用例

|  |  |
| --- | --- |
| 用例名称 | 图片信息修改 |
| 目的 | 测试图片信息修改 |
| 前提 | 进入图片详情页 |
| 测试步骤 | 1)点击地点标签设置  2)输入或更改地点、人物、AI识别信息 |
| 预期结果 | 点击确定后返回相册详情页，图片详情已刷新 |
| 实际结果 | 实际结果同预期结果一致 |

图片删除测试用例如表5-10所示。

表5-10图片删除测试用例

|  |  |
| --- | --- |
| 用例名称 | 图片删除 |
| 目的 | 测试图片信息修改 |
| 前提 | 进入图片详情页 |
| 测试步骤 | 1. 点击删除   2)弹窗询问是否确定删除，再次点击确定 |
| 预期结果 | 成功删除图片，返回相册详情页 |
| 实际结果 | 实际结果同预期结果一致 |

### 5.2.9查看分类结果

查看分类测试用例如表5-11所示。

表5-11 查看分类测试用例

|  |  |
| --- | --- |
| 用例名称 | 查看分类 |
| 目的 | 测试分类查看 |
| 前提 | 用户登录小程序 |
| 测试步骤 | 1. 打开智能分类页面 2. 点击某一分类 |
| 预期结果 | 跳转到新的页面，自动显示该用户云端中全部相同标签值的图片 |
| 实际结果 | 实际结果同预期结果一致 |

## 5.3性能测试

测试在不同网络情况下小程序的流畅度，使用多个测试账号同时登录小程序模拟使用，多个账号小程序均流畅运行，无明显卡顿现象，图片上传和加载速度可以接受，满足性能要求。

# 第6章 总结与展望

## 6.1总结

现代科技的发展，使得以往专业的摄像领域的门槛不断降低，如今只需一台轻便的手机，就可以拍摄记录所见所闻，并通过微信等社交app与朋友分享。手机上软件占用的空间不断增加，将照片存储到云端，不仅可以减少手机内存占用，并保障了数据的安全。针对以上场景，本文设计实现的旅游云相册，可以让用户将图片保存到云端，并设置为共享相册和朋友分享。

随着智能手机的发展和微信的普及，微信小程序无疑时最好的载体。在设计并实现系统的过程中，按照面向对象的软件开发思维，依次完成了系统需求分析、架构搭建与设计、系统实现和系统测试。

在毕业设计的实现过程中，我认识了解到微信小程序的开发过程，对WXML、WXSS等小程序相关的基础知识有了更深入的了解，学习并熟悉SpringBoot框架及IntelliJ IDEA等一系列相关工具的使用，了解云服务器和域名备案相关的知识，熟悉第三方接口的使用流程。

在完成毕业设计和论文的过程中，遇到了很多困难，需要去网络上查阅资料，也要去学习了解在学校里，课本上所没有的新知识，自主学习能力有了很大的提高。

## 6.2展望

由于时间和个人水平的限制，本项目还有许多不足和可以完善的地方，主要有以下几个方面：

1）系统可以加入回收站功能，方便用户恢复相册和图片。

2) 共享相册模块可以加入游客评论功能，方便网友分享交流。

3) 目前图像识别是将图片发送到第三方接口来完成，可以考虑在服务器加入图像识别模块，实现在服务器端的识别，这样，可以针对性的对图像识别算法做出优化，同时，避免图片发送第三方过程中可能发生的数据泄露问题。

4）对于用户上传的图片进行MD5编码，对于相同的图片，避免多次重复保存，节省存储空间。

# 参考文献

[1]Fangxuan (Sam) Li. Understanding Chinese tourists' motivations of sharing travel photos in WeChat[J]. Tourism Management Perspectives, 2020, 33

[2]张艳慧,初雨晴,常宴宁.浅谈微信小程序的发展与未来[J].电脑迷,2018(06):35.

[3]Lei Hao,Fucheng Wan,Ning Ma,Yicheng Wang. Analysis of the Development of WeChat Mini Program[J]. Journal of Physics: Conference Series,2018,1087(6).

[4]刘红卫.微信小程序应用探析[J].无线互联科技,2016(23):11-12+40.

[5]臧芳.基于安卓移动云相册的实现研究[J].信息与电脑(理论版),2016(06):74-75.

[6]李海航. 基于安卓相机的照片分享平台的设计与实现[D].大连理工大学,2015.

[7]王华旭. 基于Android平台的智能手机相册的设计与实现[D].中山大学,2014.

[8]许婧. 基于云存储的图片管理系统移动端的设计与实现[D].北京交通大学,2018.

[9]Zexin Huang, Botao Liu, Ying Zhong, Wei Wang, Jing An, Zijie Jiang. Data Storage Analysis of Online Shopping Cart Based on WeChat Mini Program[J]. International Core Journal of Engineering,2021,7(1).

[10]葛萌,黄素萍,欧阳宏基.基于Spring MVC框架的Java Web应用[J].计算机与现代化,2018(08):97-101.

[11]薛峰,梁锋,徐书勋,王彪任.基于Spring MVC框架的Web研究与应用[J].合肥工业大学学报(自然科学版),2012,35(03):337-340.

[12]胡启敏,薛锦云,钟林辉.基于Spring框架的轻量级J2EE架构与应用[J].计算机工程与应用,2008(05):115-118+133.

[13]徐雯,高建华.基于Spring MVC及MyBatis的Web应用框架研究[J].微型电脑应用,2012,28(07):1-4+10.

[14] 微信小程序官方文档[EB/OL]. http://mp.weixin.qq.com/wiki

[15] 科夫勒. MySQL 5 权威指南[M]. 北京: 人民邮电出版社, 2006:9~12

[16] 应用程序接口.百度百科[EB/OL]. http://baike.baidu.com/view/592964.htm

[17] 微信小程序的设计与开发[J]， 孙月玲，科技创新导报. 2018(10):200-207.

[18]刘玉佳.微信“小程序”开发的系统实现及前景分析[J].信息通信,2017(01):260-261.

[19]王婷婷.微信小程序开发[J].信息技术与信息化,2018(12):62-63.

[20]张晓燕.微信“小程序”开发的系统实现及前景[J].电子技术与软件工程,2018(12):49-50.

[21]李哲,周灵.微信小程序的架构与开发浅析[J].福建电脑,2019,35(12):66-69.

[22] 微信小程序官方文档[EB/OL]. http://mp.weixin.qq.com/wiki

# 附录 程序代码

## 附录一.小程序

### 主要函数：

getcode:function(){

var that=this;

wx.login({

success (res) {

that.setData({

code: res.code

})

console.log("code:"+that.data.code)

that.getopenid()

}

})

},

getopenid:function(){

var that=this

wx.request({

url: url+'Get/OpenId',

method:'GET',

data: {

code:this.data.code

},

success(res){

that.setData({

openid: res.data.openid

})

console.log("openid:"+res.data.openid)

that.getuser()

}

})

},

getuser:function(){

var that=this

wx.request({

url:url+'Get/User',

method:'GET',

data: {

openid:that.data.openid

},

success(res){

that.setData({

user:res.data

})

app.globalData.user=that.data.user

console.log(res.data)

}

})

},

getsharealbum:function(){

var that=this

wx.request({

url: url+'Get/ShareAlbum',

success(res){

console.log(res.data)

that.setData({

sharealbumlist:res.data

})

}

})

},

openalbum:function(item){

console.log(item.target.id)

wx.navigateTo({

url: '/pages/function/openalbum/openalbum?sharetag=true&albumid='+item.target.id

})

},

})，

newimages:function(){ //图片上传函数

var that = this;

wx.chooseImage({

sizeType: ['original', 'compressed'],

sourceType: ['album', 'camera'],

success: function (res){

that.setData({

ImageTemPath: res.tempFilePaths

})

wx.cloud.init()

for(let i = 0 ; i <that.data.ImageTemPath.length ; i++){

wx.getImageInfo({

src: that.data.ImageTemPath[i],

success(res){

that.setData({

['type['+ i +']']:res.type,

['imagename['+i+']']:new Date().getTime() +"-"+ Math.floor(Math.random() \* 1000)+'.'+res.type

})

wx.cloud.uploadFile({

cloudPath: "images/photos/"+app.globalData.user.userid+'/'+that.data.imagename[i],

filePath: that.data.ImageTemPath[i], // 小程序临时文件路径

success: res => {

console.log(res.fileID)

that.setData({

['fileID['+i+']']:res.fileID,

})

wx.getFileSystemManager().readFile({

filePath: that.data.ImageTemPath[i],

encoding:'base64',

success: res=>{

that.setData({

['base64['+i+']']:res.data,

})

Const detecturl="https://aip.baidubce.com/rest/2.0/image- classify/v2/advanced\_general?access\_token="+that.data.baidutoken;

wx.request({

url: detecturl,

data:{

image:that.data.base64[i]

},

method:"POST",

dataType:'json',

header:{

'content-type': 'application/x-www-form-urlencoded'

},

success(res){

console.log('识别图像成功')

console.log(res.data.result[0].name)

that.setData({

['resuledata['+i+']']:res.data,

['ai['+i+']']:res.data.result[0].keyword

})

wx.request({

url: url+'Upload/Image',

method:'GET',

data:{

albumid:that.data.albumid,

imagename:that.data.imagename[i],

path:that.data.fileID[i],

ai:that.data.ai[i]

},

success(){

console.log("第"+i+"张上传成功")

that.onShow()

}

})

},

})

}

})

}

})

}

})

}

}

})

},

getbaidutaken:function(){

const tokenurl='https://aip.baidubce.com/oauth/2.0/token?grant\_type=client\_credentials&clien t\_id='+app.globalData.ApiKey+'&client\_secret='+app.globalData.SecretKey;

var that=this;

wx.request({

url: tokenurl,

method:'POST',

dataType:'json',

header:{

'content-type':'application/json;'

},

success (res) {

console.log(res.data);

that.setData({

baidutoken:res.data.access\_token

})

},

fail (res){console.log("失败",res.data);}

})

},

deletealbum:function(){

var that=this

wx.showModal({

title: '提示',

content: '确定删除？',

success: function (res) {

if (res.confirm) { //这里是点击了确定以后

console.log('用户点击确定')

wx.request({

url: url+'Set/DeleteAlbum',

method:'GET',

data: {

albumid:that.data.albumid

},

success(res){

console.log(res.data)

wx.navigateBack({

delta: 0,

})

}

})

} else { //这里是点击了取消以后

console.log('用户点击取消')

}}

})

},

downloadimage:function(){

var that=this

wx.getSetting({

success(res) {

if (!res.authSetting['scope.writePhotosAlbum']) {

wx.authorize({

scope: 'scope.writePhotosAlbum',

})

}

}

})

wx.cloud.downloadFile({

fileID:this.data.image.path,

success:res=>{

console.log("下载成功")

that.saveimage(res.tempFilePath)

}

})

},

saveimage(imgurl){

wx.saveImageToPhotosAlbum({

filePath: imgurl,

success(){

console.log("保存成功")

wx.showToast({

title: '下载成功',

icon: 'success',

duration: 1000//持续的时间

})

},

fail(){

console.log("保存失败")

}

})

}

页面展示设计：  
<view class="showed-item-container">

<view wx:for="{{albumlist}}" class="showed-item">

<image id="{{item.albumid}}" src="{{item.coverimagepath}}" class="item-img" bindtap="openalbum" mode="aspectFill"></image>

<view class="text-container">

<text class="item-text">{{item.albumname}}</text>

</view>

</view>

</view>

<view class="showed-item-container">

<view wx:for="{{locationlist}}" class="showed-item">

<image id="{{item.location}}" mode="aspectFill" src="{{item.path}}" class="item-img" bindtap="openlocation"></image>

<view class="text-container">

<text class="item-text">{{item.location}}</text>

</view>

</view>

</view>

## 附录二.后端代码

### GetController

@RestController

@RequestMapping("/Get")

public class GetController {

@Autowired

private GetService getService;

@GetMapping(value = "/OpenId")

public String GetOpenId(@RequestParam(value = "code") String code) throws Exception {

System.out.println("code:"+code);

String appid="wx5d6a6caf7c8b5b67";

String secret="c4758627a56e1fe7b8454e3c408ab606";

String url = "https://api.weixin.qq.com/sns/jscode2session?appid={appid}&secret={secret}&

js\_code={code}&grant\_type=authorization\_code";

RestTemplate restTemplate = new RestTemplate();

Map<String,String> map = new HashMap<String,String>();

map.put("appid",appid);

map.put("secret",secret);

map.put("code",code);

String res = restTemplate.getForObject(url,String.class,map);

System.out.println(res);

return res;

}

@GetMapping(value = "/User")

public User GetUser(@RequestParam(value = "openid") String openid) throws Exception {

System.out.println("用户查询中");

return getService.getuser(openid);

}

@GetMapping(value = "/Albums")//根据用户id查询该用户所拥有的相册

public List<Album> GetAlbums(@RequestParam(value = "userid") String userid) throws Exception {

System.out.println("用户相册查询中");

return getService.getablums(userid);

}

@GetMapping(value = "/Albums")//查询共享相册

public List<Album> GetShareAlbum() throws Exception {

System.out.println("共享相册查询中");

return getService.getshareablum();

}

@GetMapping(value = "/Album")//根据相册id查询相册信息

public Album GetAlbum(@RequestParam(value = "albumid") String albumid) throws Exception {

System.out.println("相册查询中");

return getService.getablum(albumid);

}

@GetMapping(value = "/Images")

public List<Image> GetImages(@RequestParam(value = "albumid") String albumid) throws Exception {

System.out.println("相册图片查询中");

return getService.getimages(albumid);

}

@GetMapping(value = "/Image")

public Image GetImage(@RequestParam(value = "imageid") String imageid) throws Exception {

System.out.println("图片查询中");

return getService.getimage(imageid);

}

@GetMapping(value = "/Search")

public List<Image> Search(@RequestParam(value = "label") String label,

@RequestParam(value = "ai") String ai,

@RequestParam(value = "location") String location,

@RequestParam(value = "userid") String userid) throws Exception {

System.out.println("搜索中"+label+ai+location +userid);

return getService.search(label,ai,location,userid);

}

@GetMapping(value = "/ImagesByLocation")

public List<Image> GetImagesByLocation(@RequestParam(value = "userid") String userid,@RequestParam(value = "location") String location)throws Exception{

System.out.println("查询该地点所有的图片中");

return getService.getimagesbylocation(location,userid);

}

@GetMapping(value = "/Locations")

public List<Image> GetLocations(@RequestParam(value= "userid") Stringuserid)throws Exception{

System.out.println("查询所有地点中");

return getService.getlocations(userid);

}

}

### GetService：

@Service

public class SetService {

@Autowired

private SetMapper setmapper;

public int deleteimage(String imageid){

return setmapper.deleteimage(imageid);

}

public int deletealbum(String albumid){

return setmapper.deletealbum(albumid);

}

public int updatealbum(String albumtype,String description,String albumid,String albumname){

return setmapper.updatealbum(albumtype,description,albumid,albumname);

}

public int updateimage(String imageid,String label,String ai,String location){

return setmapper.updateimage(imageid,label,ai,location);

}

public int updateuser(String name,String phonenumber,String userid){

return setmapper.updateuser(name,phonenumber,userid);

}

public int coverimage(String albumid, String coverimagepath) {

return setmapper.coverimage(albumid,coverimagepath);

}

}

### GetMapper

@Mapper

public interface SetMapper {

@Update("UPDATE image SET status='delete' WHERE imageid=#{imageid}")

int deleteimage(@Param("imageid") String imageid);

@Update("UPDATE album SET status='delete' WHERE albumid=#{albumid}")

int deletealbum(@Param("albumid") String albumid);

@Update("UPDATE image SET label=#{label},ai=#{ai},location=#{location} WHERE imageid=#{imageid}")

int updateimage(@Param("imageid") String imageid,@Param("label") String label,@Param("ai") String ai,@Param("location") String location);

@Update("UPDATE album SET albumtype=#{albumtype},description=#{description},albumname=#{albumname} WHERE albumid=#{albumid}")

int updatealbum(@Param("albumtype") String albumtype,@Param("description") String description,@Param("albumid") String albumid,@Param("albumname")String albumname);

@Update("UPDATE user SET name=#{name},phonenumber=#{phonenumber} where userid=#{userid}")int updateuser(@Param("name") String name,@Param("phonenumber") String phonenumber,@Param("userid") String userid);

@Update("UPDATE album SET coverimagepath=#{coverimagepath} WHERE albumid=#{albumid}")

int coverimage(String albumid, String coverimagepath);

}