**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是最好的选择。

设计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，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、系统测试的顺序来进行阐述。

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 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；Image Recognition；Cloud Album；

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](#_Toc69252431)

[1.1 研究的背景 1](#_Toc69252432)

[1.2 研究的目的和意义 1](#_Toc69252433)

[1.2.1 研究目的 1](#_Toc69252434)

[1.2.2 研究意义 2](#_Toc69252435)

[1.3国内外技术现状分析 2](#_Toc69252436)

[1.4主要工作 3](#_Toc69252437)

[1.4.1 前端 3](#_Toc69252438)

[1.4.2 后端 3](#_Toc69252439)

[1.4.3 云服务部署 3](#_Toc69252440)

[第2章 相关概念与技术介绍 4](#_Toc69252441)

[2.1 相关概念 4](#_Toc69252442)

[2.1.1 微信小程序 4](#_Toc69252443)

[2.1.2 阿里云 4](#_Toc69252444)

[2.1.3 Java 4](#_Toc69252445)

[2.1.4 MySQL 5](#_Toc69252446)

[2.2 相关技术及工具 5](#_Toc69252447)

[2.2.1前后端分离 5](#_Toc69252448)

[2.2.2 Postman 5](#_Toc69252449)

[2.2.3 Xshell 5](#_Toc69252450)

[2.2.4 Git 6](#_Toc69252451)

[2.2.5 Navicat for MySQL 6](#_Toc69252452)

[第3章 系统分析与设计 7](#_Toc69252453)

[3.1 可行性分析 7](#_Toc69252454)

[3.1.1经济可行性 7](#_Toc69252455)

[3.1.2技术可行性 7](#_Toc69252456)

[3.2 需求分析 8](#_Toc69252457)

[3.1.1 功能需求 8](#_Toc69252458)

[3.1.2 非功能性需求 9](#_Toc69252459)

[3.1.3 性能需求 9](#_Toc69252460)

[3.3功能分析 10](#_Toc69252461)

[3.3.1总体功能 10](#_Toc69252462)

[3.3.2用户模块 10](#_Toc69252463)

[3.3.3相册模块 11](#_Toc69252464)

[3.3.4图片模块 11](#_Toc69252465)

[3.3.5搜索模块 12](#_Toc69252466)

[3.4软件整体结构框架 13](#_Toc69252467)

[3.5 数据库设计 14](#_Toc69252468)

[3.5.1 E-R图 14](#_Toc69252469)

[3.5.2 用户信息表 14](#_Toc69252470)

[3.5.3 相册信息表 15](#_Toc69252471)

[3.5.4 图片信息表 16](#_Toc69252472)

[第4章 系统实现 18](#_Toc69252473)

[4.1 开发环境配置 18](#_Toc69252474)

[4.1.1微信小程序注册及开发工具安装 18](#_Toc69252475)

[4.1.2云服务器购买及配置 18](#_Toc69252476)

[4.1.3百度图像识别接口申请 19](#_Toc69252477)

[4.1.3软件版本 19](#_Toc69252478)

[4.2小程序开发 19](#_Toc69252479)

[4.2.1.页面设计 19](#_Toc69252480)

[4.2.1登录及权限获取 22](#_Toc69252481)

[4.2.2图片上传 23](#_Toc69252482)

[4.2.3搜索功能 24](#_Toc69252483)

[4.2.3相册及图片浏览 25](#_Toc69252484)

[4.2.4查看智能分类 27](#_Toc69252485)

[4.3后端开发 28](#_Toc69252486)

[4.4项目部署 30](#_Toc69252487)

[第5章 系统测试 31](#_Toc69252488)

[5.1测试环境 31](#_Toc69252489)

[5.2测试用例 32](#_Toc69252490)

[5.2.1用户登录 32](#_Toc69252491)

[5.2.2个人相册浏览 32](#_Toc69252492)

[5.2.3共享相册浏览 33](#_Toc69252493)

[5.2.4新建相册 33](#_Toc69252494)

[5.2.5相册设置 34](#_Toc69252495)

[5.2.6上传图片 34](#_Toc69252496)

[5.2.7下载图片 35](#_Toc69252497)

[5.2.8图片信息修改 35](#_Toc69252498)

[5.2.9查看分类结果 36](#_Toc69252499)

[5.3性能测试 37](#_Toc69252500)

[第6章 总结与展望 38](#_Toc69252501)

[6.1总结 38](#_Toc69252502)

[6.2展望 38](#_Toc69252503)

[参考文献 40](#_Toc69252504)

[附录 程序代码 42](#_Toc69252505)

[附录一.小程序 42](#_Toc69252506)

[主要函数： 42](#_Toc69252507)

[页面展示设计： 47](#_Toc69252508)

[附录二.后端代码 48](#_Toc69252509)

[GetController 48](#_Toc69252510)

[GetService： 50](#_Toc69252511)

[GetMapper 50](#_Toc69252512)

# 第1章 绪论

## 1.1 研究的背景

随着物质水平的提高和现代科技的发展，人们的生活变得越来越便利，智能手机已经是现代人生活不可分割的一部分。在满足基本的物质需求的基础上，人们开始热衷于精神需求的满足，旅游业蓬勃发展。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，成像效果足以媲美部分专业相机。

随着旅行时用手机拍照已经成为一种习惯，手机里的照片越来越多。而现在的智能手机往往无法扩展存储空间，在手机内存有限的条件下，许多用户都会选择将自己照片存放在云端来节省手机内存。并且，人们热衷于分享照片[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需求。

## 1.2 研究的目的和意义

### 1.2.1 研究目的

作为一款云相册类产品，除了实现备份或储存照片的基础功能，还有哪些是可以改进的呢？微信作为一款手机必备软件，如果采用微信小程序的方式，用户无需下载专用APP便可使用，同时也节省了手机内存空间，得益于微信提供的各种接口，用户可以得到良好的使用体验。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。为了能够增加用户的使用时间和粘度[2]，所以还应具有分享功能，可以发布自己的旅游照片，与天南海北的网友分享。该小程序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想要的图片。将后台服务器及数据库部署到云服务器上，降低后期维护成本，同时，用户照片使用微信云开发提供的存储功能进行保存，保证用户加载图片的速度，同时管理员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

随着互联网从增量市场逐渐成为存量市场，互联网巨头们以自身拥有的超级APP为基础搭建小程序平台。据腾讯财报显示，微信月活跃用户突破 12 亿。微信小程序受到众多用户和开发者的喜爱，小程序发展前景广阔[3]，更是未来的趋势[4]。小程序以小见大[5]，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。

市面上有许多相册类应用[6]-[9]，如腾讯相册管家、一刻相册、一起传相册等，不过多数为App形式，需要用户下载软件，而且该相册类APP往往添加了过多功能，导致过于臃肿，大量占据了本就不充足的手机空间；也存在着一些基于小程序的相册类应用，但是使用起来，页面布局不够人性化，功能比较单一，通常只有简单的相册上传和分享功能，不具备图像智能识别功能，同时广告较多，用户体验较差。

因此，将臃肿的相册APP与微信小程序相结合，取长补短，保留核心的功能，利用好微信小程序提供的云存储功能 [10]，给用户带来良好的使用体验。

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，众多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。

## 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、上传和下载照片、设置相册、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[11]。

### 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Spring Boot框架开发，用户及相册和图片信息使用Mysql数据库进行保存，图片文件保存使用微信云存储。后端服务器提供用户信息查询、相册及图片信息的查询及修改、关键词搜索等功能的接口[12]。

### 1.4.3 云服务部署

使用阿里云的云服务器进行项目的部署，服务器系统采用CentOS 7.3，进行域名备案并将之解析到云服务器的IP地址，通过域名访问服务器。

# 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

微信小程序无需下载和安装，只要通过扫描二维码或搜索就可打开。对于开发者来说，微信小程序的开发门槛相对较低，难度比APP小。它能满足简单基础应用的需求，微信小程序可以实现许多功能。小程序提供了一个简单高效的应用程序开发框架、丰富的组件和API，可以帮助开发者更好的开发小程序。而且依托于微信庞大的用户群体，小程序的市场前景十分广阔。[13]-[17]

### 2.1.2 阿里云

目前全球云计算市场集中度高，亚马逊云占据了全球云服务市场的主要份额[18]，国内则以阿里云为主[19]。云服务器是阿里云提供的基础云计算服务。使用ECS，不同于本地主机，ECS可以根据业务需要随时扩充或减少服务器资源，如CPU、磁盘和带宽等，按量付费。通过云服务器管理控制台来进行云服务器的各项配置。

云服务器与本地主机主机一样，需要安装操作系统，Linux作为多用户、多任务、多线程及多CPU的开源操作系统，它免费使用、自由传播，受到广大开发者的喜好，因此，选择Linux的发行版CentOS来作为服务器的系统。

### 2.1.3 Java

Java具有面向对象、简单、安全和跨平台等特点，是许多应用系统的理想开发语言，是业界最流行的开发语言之一，本系统的后端服务器采用Java编写。

IDEA是一款Java的集成环境，在业界被公认为最好的Java开发工具之一。许多公司也使用IDEA进行项目的开发，IDEA的强大功能，可以极大提高开发效率。本项目使用IDEA进行后端服务器的开发。

本项目使用Spring Boot框架，Spring Boot基于Spring设计，继承了Spring框架原有的优秀特性，而且进一步简化了应用的搭建和开发过程，使用Spring Boot框架，可以快速便捷的建立Java项目[20]-[23]。

### 2.1.4 MySQL

MySQL 是最流行的关系型数据库，它体积小、速度快，而且作为一款开源数据库，无疑是本项目的最好选择[24]。

## 2.2 相关技术及工具

### 2.2.1前后端分离

随着互联网的快速发展，前后端分离已经成为互联网项目开发的行业标准方式。前端和后端独立开发，服务器独立部署。前后端分离提高了工作效率[25]，分工明确，提高了性能，降低了维护成本，前端通过get或post在浏览器和web服务器之间进行异步数据传输，从而与服务器进行交互。

### 2.2.2 Postman

接口测试是项目开发中的重要内容，利用接口测试可以有效降低测试成本[26]，开放API存在安全问题，需要进行加密处理，可使用HTTPS进行加密传输[27]。Postman是一个接口测试工具，它可以有效地帮助后端开发人员进行接口测试。

### 2.2.3 Xshell

Xshell是一个功能强大的安全终端软件，支持多种协议。使用Xshell可以方便的配置和管理云服务器。

### 2.2.4 Git

Git是一种开源的分布式版本控制软件[28]，用于快速有效地处理从小到大的项目的版本管理。GitHub是一个基于Git的远程文件托管平台，通过使用GitHub，可以方便快捷的进行项目的协同开发和版本管理[29]。

### 2.2.5 Navicat for MySQL

Navicat for MySQL是一款管理和开发MySQL的软件[30]，可连接多个MySQL数据库。提供了直观而强大的图形界面，给开发者提供了很大便利。

# 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，在软件开发前进行可行性分析，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，可以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 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，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。

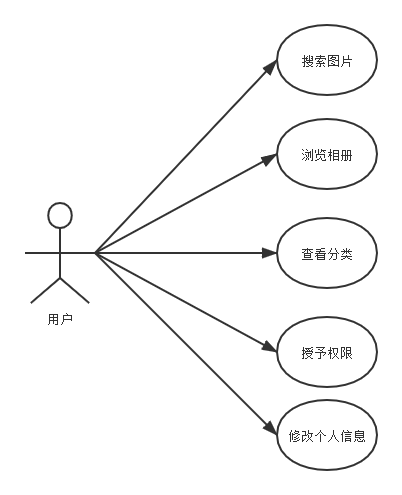
用户用例图如图3-1、图3-2、图3-3所示。

图3-1 用例图1

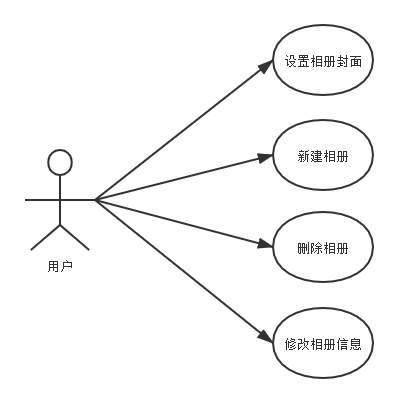
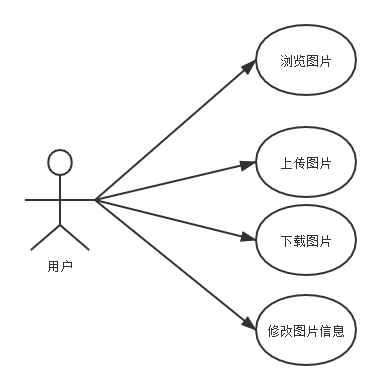


图3-2 用例图2 图3-3 用例图3

### 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：

1) 易用性，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。

2) 可靠性，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，避免本地服务器可能发生的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压力。

3) 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。

### 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，图片实际存储于微信云存储上，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请求，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3-4所示。

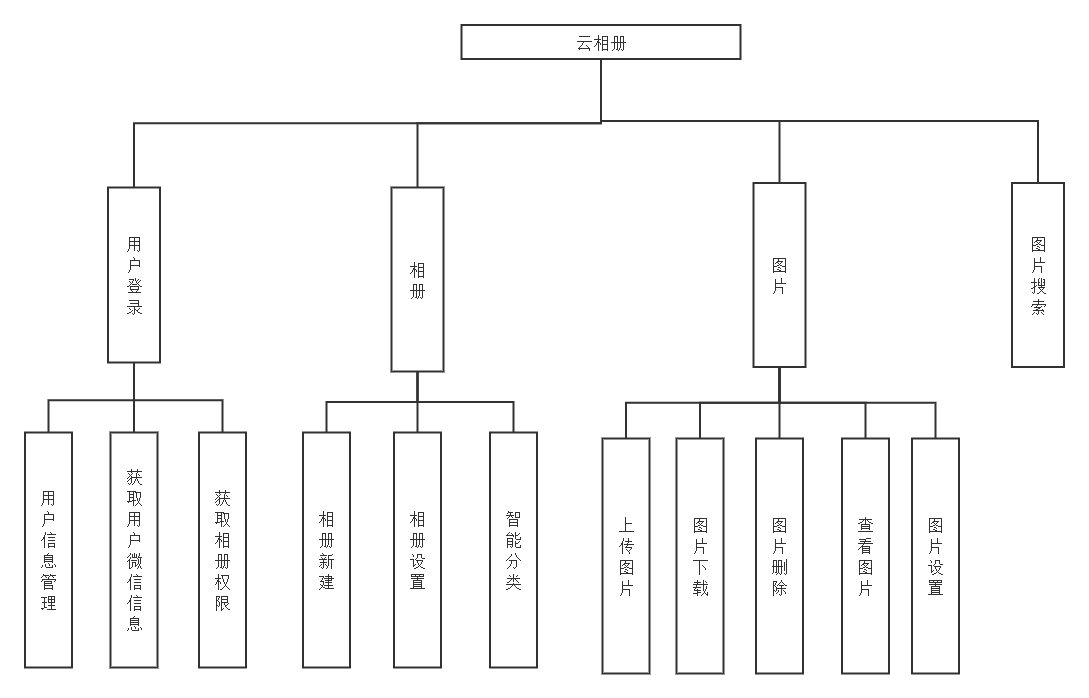


图3-4 系统功能图

### 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。

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数据库中，如果是已注册用户，则自动展示用户个人相册。

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，如果已经获取到相册权限，则会自动跳过该过程。

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、修改相册信息、删除相册、上传相册图片。

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。

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。

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时，可删除相册。

智能分类：小程序会根据图像识别的结果以及用户自定义的标签，对用户的图片进行分类，将相同标签或识别结果的图片归为同一类。

### 3.3.4图片模块

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和浏览。上传图片时的流程如图3-5所示。

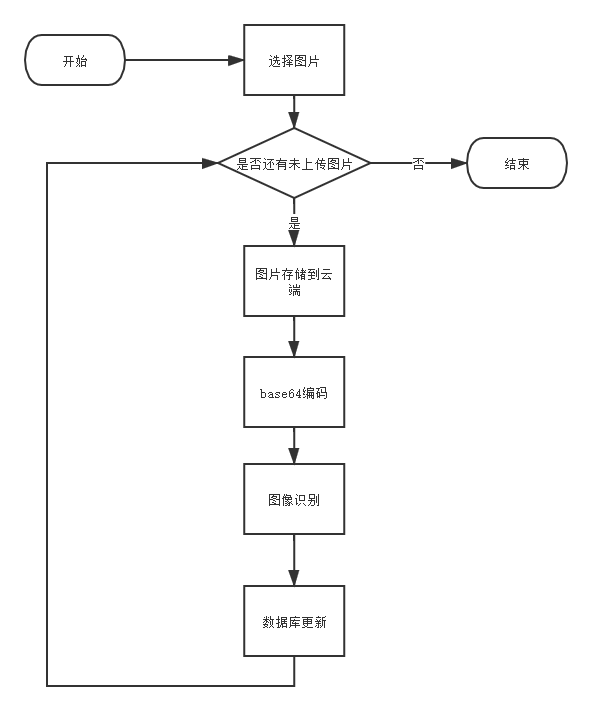


图3-5 图片上传流程图

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

图片删除：用户可删除个人相册中的图片

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

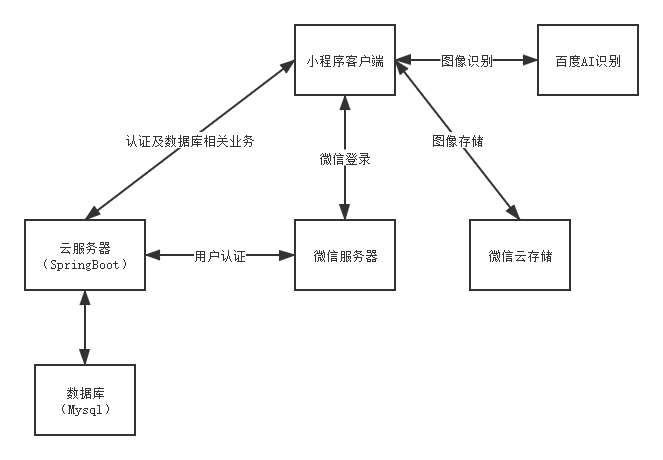
### 3.3.5搜索模块

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

本应用基于微信小程序，采用前后端分离的思想进行开发，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，数据库采用MySQL。

当用户打开小程序的时候，小程序会自动获取用户的code，再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，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。软件整体结构如图3-6所示。

图3-6 整体结构图

## 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、图片表，实体关系如图3-5所示。

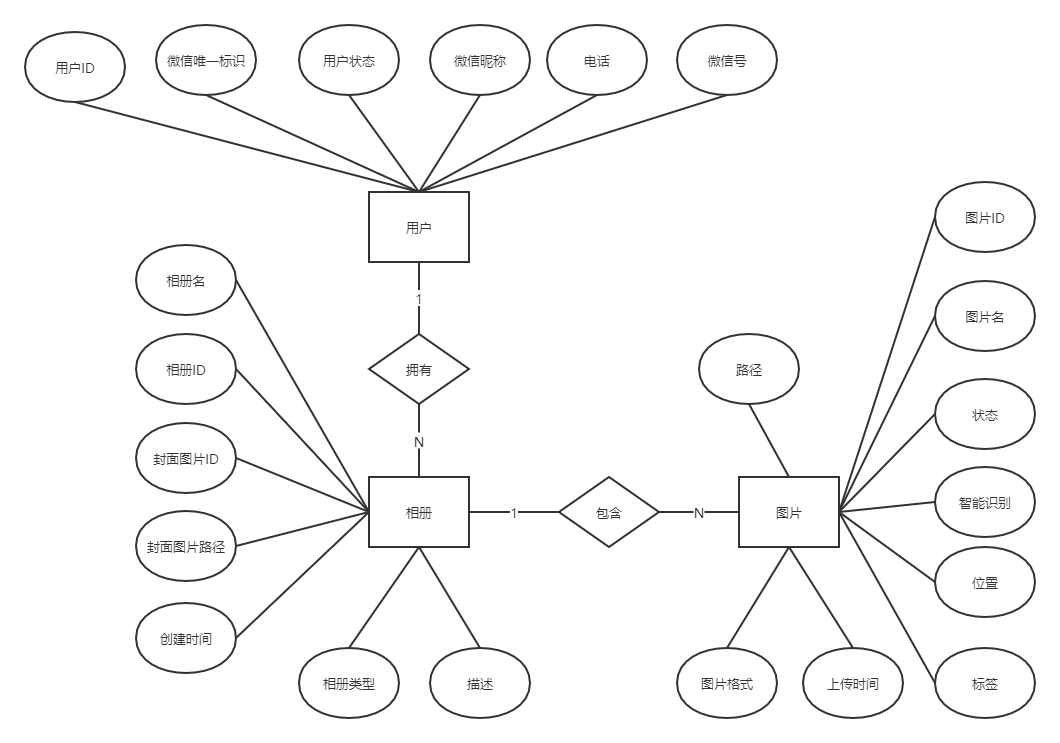


图3-5 E-R图

### 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Spring Boot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：

Mysql数据库；

Xshell，便于远程控制操作云服务器；

Postman，进行接口调试；

Navicat 12 for MySQL，管理维护数据库；

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，本项目购买的是轻量应用服务器，足够使用，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证书并进行配置，全部完成后，小程序可以通过访问<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图像识别接口申请

通过调用百度图像识别接口来进行图像识别，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

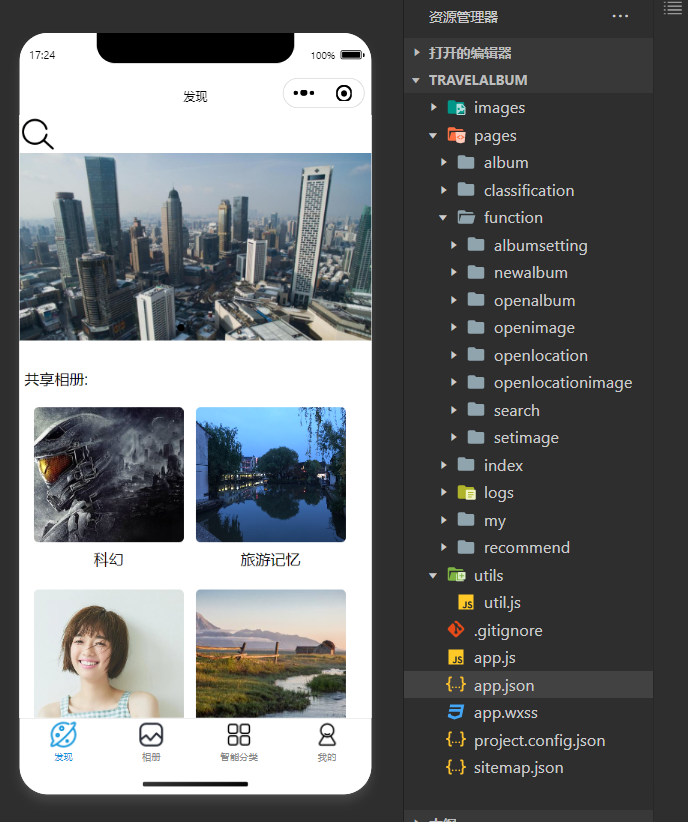
表4-1开发环境配置表

|  |  |
| --- | --- |
| 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 |
| Spring Boot | 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

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。将临时路径用this.setData( )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，用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拼接成图片名，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https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。

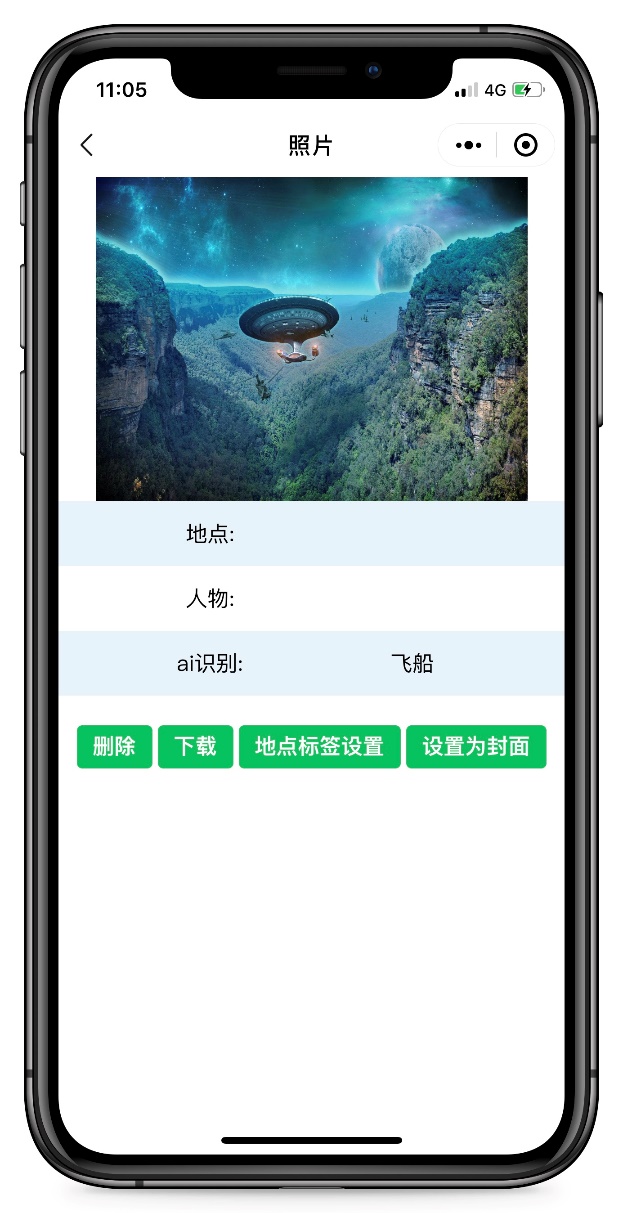
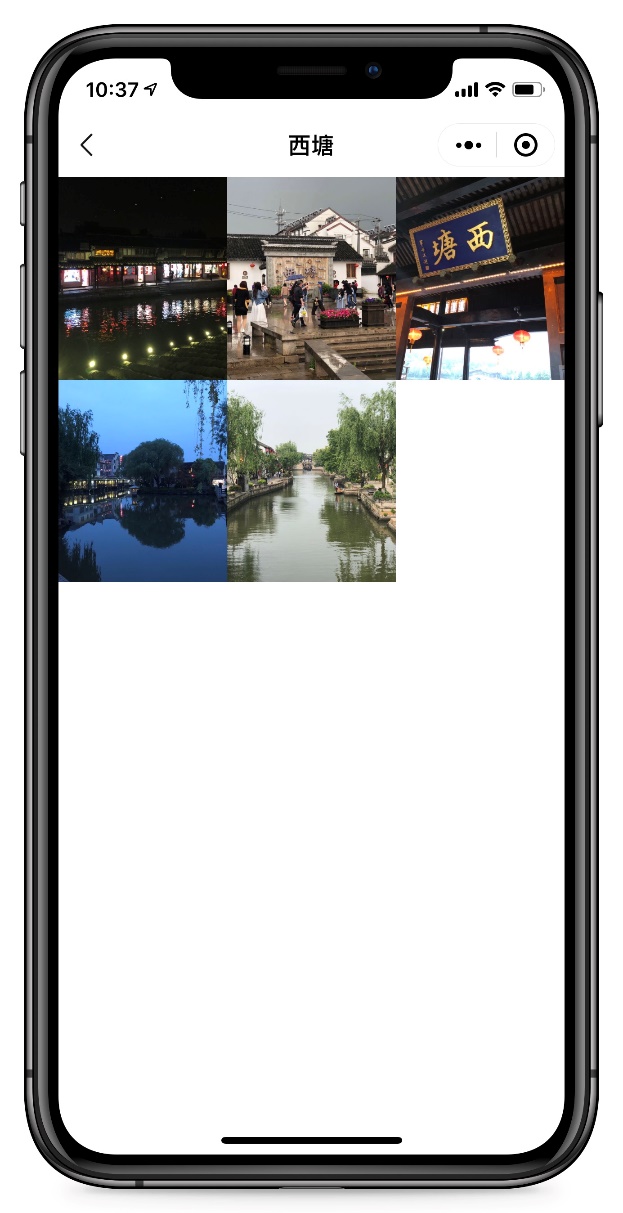
从共享相册打开图片和从个人相册打开图片后跳转的图片详情界面会根据sharetag自动对相关按钮进行隐藏，保证用户相册安全。打开图片如图4-13、图4-14所示。

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，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，而且保障了数据的安全。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，微信小程序无疑是本项目的最好载体。在设计并实现系统的过程中，按照软件开发过程，依次完成了需求分析、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，我进一步熟悉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，学习并熟悉了Spring Boot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，需要自行去网络上查阅资料，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进步。

## 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. 关于相册类产品的分析和思考[EB/OL]. https://www.toutiao.com/i6420627310398931457
3. 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.
4. 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).
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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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).
11. 微信小程序官方文档[EB/OL]. http://mp.weixin.qq.com/wiki
12. 应用程序接口.百度百科[EB/OL]. http://baike.baidu.com/view/592964.htm
13. 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.
14. 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.
15. 王婷婷.微信小程序开发[J].信息技术与信息化,2018(12):62-63.
16. 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.
17. 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.
18. 郭朝先,胡雨朦.中外云计算产业发展形势与比较[J].经济与管理,2019,33(02):86-92.
19. 于斌.横向评测阿里云、百度云、腾讯云和华为云[J].大数据时代,2019(08):12-21.
20. 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.
21. 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.
22. 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.
23. 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.
24. 科夫勒. MySQL 5 权威指南[M]. 北京: 人民邮电出版社, 2006:9~12
25. 孙思杰.Web项目基于前后端分离模式的设计与应用[J].科技创新与应用,2020(27):96-97.
26. 何琼月.软件测试中接口测试概述与实践[J].电子测试,2021(02):80-81+75.
27. 刘俊.软件开发开放API接口的安全处理[J].信息与电脑(理论版),2017(08):201-203.
28. 王真.版本控制工具在软件开发项目管理中的应用——以GIT为例[J].项目管理技术,2020,18(06):131-134.
29. 秦佳.软件配置管理中版本控制的研究[J].软件,2019,40(03):137-139.
30. 赵光亮,舒小松.Navicat for MySQL平台中的SQL语言分析与应用[J].无线互联科技,2017(19):74-75.

# 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);

}