四川工程职业技术学院

秀视频

学号：201612032152

姓名：徐塬峰

专业：物联网应用技术

指导教师：曹冰

成绩：

完成时间：

摘 要

如今，短视频社交已经成为当前最热门的一种社交方式，基于安卓或者苹果平台开发的短视频社交软件依然存在一些问题，例如占用用户的内存，各个平台之间无法跨越平台使用，安全性能不高等问题。

然而在微信小程序出现的今天，我们可以利用微信小程序来解决这个问题，因此，本文旨在为企业开发一款基于微信小程序的短视频社交平台。

本文主要是针对微信小程序开发，后台管理系统的开发，采用计算机软件技术，基于SpringBoot微服务框架、SpringCloud分布式技术建立一个微信小程序的短视频平台，主要实现的功能是用户对短视频的发布，编辑，点赞，分享，评论，浏览。管理员对视频的管理，专栏的管理，音乐的管理等功能。本文首先对该平台进行需求分析，可行性分析，功能需求分析，其次通过需求分析、总体方案设计来确定该平台各个模块以及后台管理系统的方案，紧接着确定好的功进行编码实现，最后整合所有功能进行功能测试。本项目中，我主要负责了后端开发，架构设计，程序部署。

关键字： java、springboot、软件架构、微服务

目 录

[第一章 绪论 1](#_Toc533592693)

[1.1 研究背景 1](#_Toc533592694)

[1.2 研究目的及意义 1](#_Toc533592695)

[1.3 研究的内容 1](#_Toc533592696)

[第二章 系统总体方案设计 3](#_Toc533592697)

[2.1 可行性分析 3](#_Toc533592698)

[2.2 系统架构 3](#_Toc533592699)

[2.2.1 微服务 3](#_Toc533592700)

[2.2.2 框架 4](#_Toc533592701)

[2.3 总体功能 5](#_Toc533592702)

[2.3.1 微信小程序端 5](#_Toc533592703)

[2.3.2 后台管理系统 6](#_Toc533592704)

[2.4 技术选型 6](#_Toc533592705)

[2.4.1 小程序端 7](#_Toc533592706)

[2.4.2 后台管理系统端 7](#_Toc533592707)

[2.4.3 分布式 7](#_Toc533592708)

[2.4.4 开发环境 7](#_Toc533592709)

[2.5 关键技术介绍 8](#_Toc533592710)

[2.5.1 SpringBoot框架 8](#_Toc533592711)

[2.5.2 SprinMVC框架 8](#_Toc533592712)

[2.5.3 Mybatis框架 9](#_Toc533592713)

[2.5.4 SpringCloud框架 9](#_Toc533592714)

[第三章 软件系统设计 11](#_Toc533592715)

[3.1 项目结构设计 11](#_Toc533592716)

[3.2 界面设计 12](#_Toc533592717)

[3.2.1小程序端界面设计 12](#_Toc533592718)

[3.2.2 后台管理系统界面设计 14](#_Toc533592719)

[3.3 功能层设计 15](#_Toc533592720)

[3.3.1 小程序端功能设计 16](#_Toc533592721)

[3.2.2 小程序端代码实现 19](#_Toc533592722)

[3.3.3 后台管理系统功能设计 20](#_Toc533592723)

[3.3.4 后台管理系统代码实现 21](#_Toc533592724)

[3.4 数据库设计 23](#_Toc533592725)

[3.4.1 数据模型设计 23](#_Toc533592726)

[3.4.2 数据层设计 24](#_Toc533592727)

[3.4.3 其他数据库设计 26](#_Toc533592728)

[第四章 系统分析与调试 28](#_Toc533592729)

[4.1测试原理 28](#_Toc533592730)

[4.1.1 单元测试 28](#_Toc533592731)

[4.1.2 功能测试 28](#_Toc533592732)

[4.1.3 接口测试 28](#_Toc533592733)

[4.2测试环境 28](#_Toc533592734)

[4.3单元测试用例 29](#_Toc533592735)

[4.4功能测试用例 29](#_Toc533592736)

[4.4.2 后台音乐管理测试用例 30](#_Toc533592737)

[4.4.3 用户登陆注册测试用例 31](#_Toc533592738)

[4.5接口测试 31](#_Toc533592739)

[4.6测试结论 32](#_Toc533592740)

[第五章 软件安装 33](#_Toc533592741)

[5.1安装准备 33](#_Toc533592742)

[5.1.1.操作系统环境 33](#_Toc533592743)

[5.1.2.集成开发环境 33](#_Toc533592744)

[5.1.3.网络环境 33](#_Toc533592745)

[5.2安装流程 33](#_Toc533592746)

[5.2.1 配置 33](#_Toc533592747)

[5.2.2 分布式组件安装 33](#_Toc533592748)

[5.2.3 小程序安装 33](#_Toc533592749)

[5.2.4 后台管理系统安装 34](#_Toc533592750)

[结论 35](#_Toc533592751)

[致谢 36](#_Toc533592752)

[参考文献 37](#_Toc533592753)

第一章 绪论

1.1 研究背景

近年来，短视频已经成为内容创业和社交平台的重要传播方式之一，其快速发展得益于移动互联网的深入普及，随着信息化时代的内容大爆炸，现在人们已经从传统的图文模式中跳脱出来，开始用短视频进行交流[1]。借助网红经济时代的网红效应短视频迅速发展成规模庞大的内容创业形式。除了作为一种内容创业形式带来巨额回报之外，其本身所具有的社交传播属性也带来了社交平台的一轮全新洗牌。短视频具有视频短、传播快、生产流程简单、制作门槛低，这些特点让短视频从出现到崛起的时间十分短暂[2]。微信小程序，是一款不需要安装即可使用的应用，它实现了“触手可及”的梦想，用户扫一扫即可使用该程序，其简单的开发方式以及其”即用即走”的使用理念，深受开发人员以及广大用户的喜爱[3]。微服务技术框架，是目前大热的技术框架，将传统的软件开发不断的细化，将复杂的软件解耦，各个微服务之间是松耦合的。每个微服务仅关注于完成一件任务并很好地完成该任务，适用于团队开发。

1.2 研究目的及意义

在以往的软件中大多都是在安卓、苹果平台上运行的，这样的软件会占用用户的手机内存，以及无法垮平台使用，安全性能不佳。在这种环境下，基于微信平台的微信小程序就体现了其优势，而我们要设计一款基于微信平台的短视频社交平台，该平台可以解决不占用用户的内存，跨平台，安全性问题并拥有短视频平台的所有功能。

1.3 研究的内容

使用微信小程序，SpringBoot微服务框架开发一款短视频社交软件。本人在课题中主要承担了秀视频的需求调研、系统分析与各模块功能设计的工作。对日常工作所遇到的一些问题进行了总结，通过面向对象的方法对系统需求做出分析。然后本着高效率工作、高质量管理的原则，对系统中涉及的功能进行了开发。最后设计了功能测试用例来检查系统各项功能的正确性和完整性。

第二章 系统总体方案设计

2.1 可行性分析

首先由于该平台开发所需的环境都能正常提供，并不需要额外购买，因此，在经济方面是可行的。其次系统以SpringBoot微服务技术为基础，选择MySQL数据库，Redis数据库，使用Eclipse集成开发工具来开发系统，经过市场调研，这些技术都已经成熟，所以，在技术方面也是可行的。最后，该系统的开发采用了前后分离式的开发方式，目前大多数软件公司都采用了前后分离的开发方式，主要是为了方便前端开发人员与后端开发人员的开发，所以开发风格也是可行的。

2.2 系统架构

2.2.1 微服务

架构选取上，我们采用了微服务架构，微服务架构是将单个的整体应用程序分割成更小的项目关联的独立的服务。一个服务通常实现一组独立的特性或功能，包含自己的业务逻辑和适配器。各个微服务之间的关联通过restFul api[9]来实现。这些独立的微服务不需要部署在同一个虚拟机，同一个系统和同一个应用服务器中。这样做的好处是由于每个服务都是独立并且微小的，由单独的团队负责，仍然可以采用敏捷开发模式，自由的选择合适的技术，甚至可以重写老服务，当然都要遵守统一的API约定[4]。在本软件中，我们使用了微服务架构中的服务发现，统一配置管理，在项目运行之后，通过分布式注册中心来获取分布式统一配置中心的坐标，获取项目相关的配置。其中架构设计如下:

其中系统主要分为四个端, 软件架构图如图2-1所示:

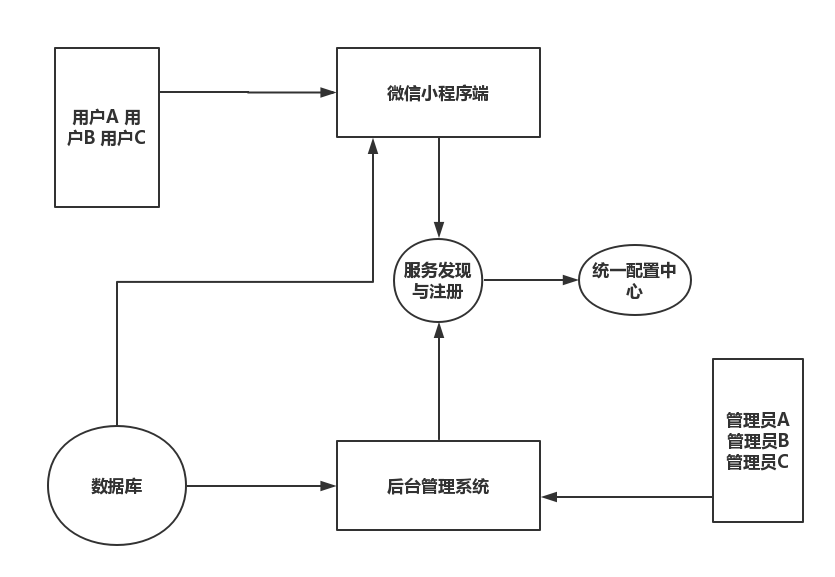


图2-1 软件架构图

2.2.2 框架

由上文可知，本系统小程序端、后台管理系统中后端均采用基于Springboot框架来进行搭建的，利用Springboot框架，可以比较好的实现模块之间的同时开发，可以构建一个符合MVC模式的Web系统，提高开发效率。在Springboot框架中，作为基本框架的SpringMVC主要是负责实现MVC（Model View Controller）的分离，负责用户请求和业务逻辑组件的交互；Mybatis提供持久化支持，负责持久化保存应用状态；Springboot中Spring主要是管理各个对象之间的关系，同时协调SpringMVC与Mybatis之间的工作关系，Springboot遵循“约定大于配置”减少了Spring配置化的开发，提高了效率。

其中框架思路如下图2-2所示，三层框架图如图2-3所示：

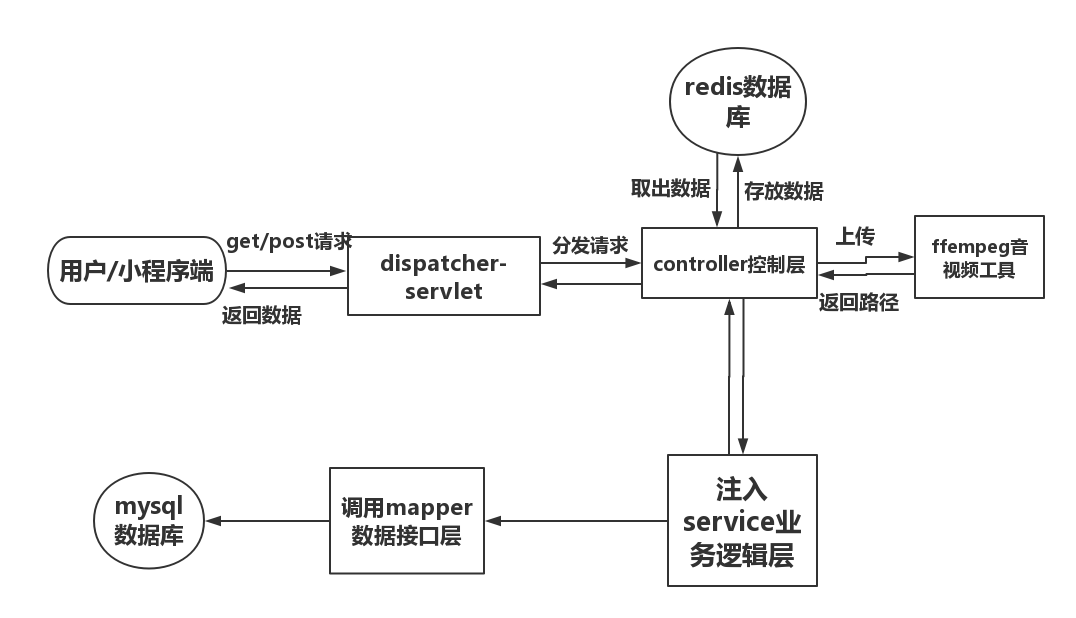


图2-2 框架思路

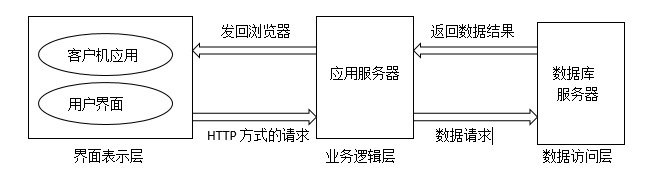


图2-3三层架构图

2.3 总体功能

2.3.1 微信小程序端

1. 用户信息模块:包含:注册、登陆、发布短视频、编辑短视频，其中编辑视频包含为视频增加背景音乐、为视频增加文字、为视频增加滤镜功能。
2. 短视频信息模块:视频详情页面、视频的点赞、评论、分享、搜索。
3. 举报模块：用户对违规视频进行举报
4. 关注用户 用户可以对视频发布者进行关注、查看它发布的短视频
5. 专栏模块 对视频进行分类，用户将视频发布到不同的专栏下。

2.3.2 后台管理系统

1. 对视频的管理，包含对被举报的视频进行下架。
2. 对背景音乐的管理
3. 管理员操作日志记录
4. 邮箱系统，包含对管理员、用户发送邮件通知
5. 对专栏的管理。
6. 数据统计模块 统计用户数量，视频数量，流量。

整体功能如图2-4 所示：

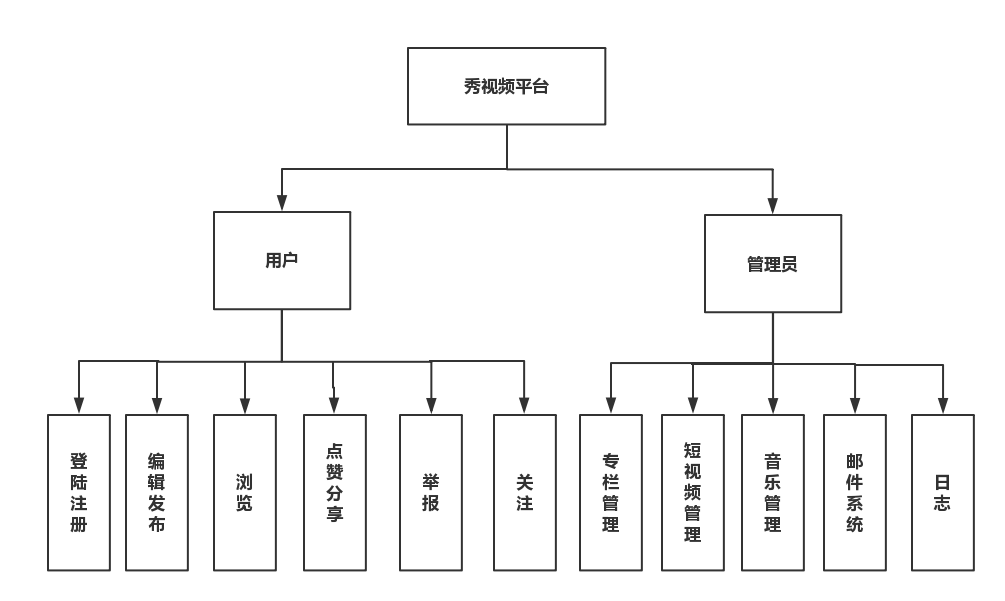


图2-4 软件功能图

2.4 技术选型

在重大产品决策或者大规模应用开发前一般需要进行技术选型，特别是需要开发一个之前没有了解过的系统或者应用时，其目的是为了降低产品研发的技术风险。所以首先需要明确为什么需要技术选型、需要达到什么目的，整个过程需要有一套的组织流程来保证。本项目是在四个端中进行开发，所以这里分别对四个端以及相关的技术选型。

2.4.1 小程序端

前端:根据小程序官方开发文档,其中使用了wxss,wxml,js相关技术。

后端:使用springboot结合spring,springmvc,mybatis，数据库选择了mysql数据库,redis数据库。

组件:使用了wxseasrch小程序开源搜索组件。

工具:音视频编辑工具选择了ffmpeg。

2.4.2 后台管理系统端

网站前端:采用了html,css,javascript ,并使用了bootsrap，jquery前端框架。

组件:bootstrap-talbe表单组件,webuploader文件上传组件。

后端:使用Springboot结合Spring,Springmvc,mybatis，数据库选择了mysql数据库,redis数据库。

2.4.3 分布式

分布式服务发现：使用springcloud eureka服务发现与注册，管理微服务中的各个微服务应用。

分布式统一配置中心：采用springcloud config 将小程序端的配置文件，后台管理系统的配置文件统一配置。

2.4.4 开发环境

项目中所采用的开发环境选择如下:

JDK1.8 java编译环境

Mysql 5.6 数据库环境

Navicat 数据库管理工具

RedisManager 数据库管理工具

MySQL Workbench 6.3 CE 数据建模工具

Tomcat8.0 服务器环境

Eclipse STS3 集成开发工具

微信小程序 集成开发工具

Webstorm 集成开发工具

Windows Server2008 操作系统

2.5 关键技术介绍

2.5.1 SpringBoot框架

[Spring Boot](http://projects.spring.io/spring-boot)是由[Pivotal](http://gopivotal.com/)团队提供的全新框架，属于spring旗下的一个项目，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，它使用“习惯优于配置”的理念，从而使开发人员不再需要定义样板化的配置。使用springboot很容易创建一个独立运行的spring项目，并且可以几乎不使用spring配置或者使用很少的配置[7]。

2.5.2 SprinMVC框架

SpringMVC是一种基于Java，实现了Web MVC设计模式，请求驱动类型的轻量级Web框架[8]，即使用了MVC架构模式的思想，将Web层进行职责解耦。基于请求驱动指的就是使用请求-响应模型，框架的目的就是帮助我们简化开发，SpringMVC也是要简化我们日常Web开发。

其SpringMVC原理图如图2-5所示

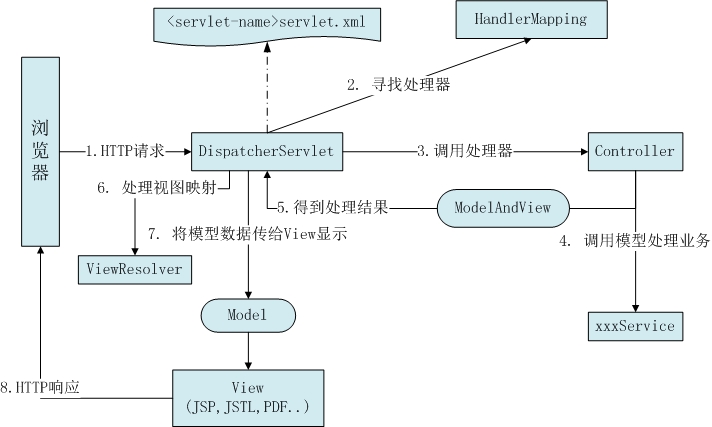


图2-5 springmvc原理图

2.5.3 Mybatis框架

MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生信息，将接口和 Java 的 POJOs(Plain Old Java Objects,普通的 Java对象)映射成数据库中的记录。

2.5.4 SpringCloud框架

Spring Cloud是一个微服务框架，相比Dubbo等RPC框架, Spring Cloud提供的全套的分布式系统解决方案。

Spring Cloud对微服务基础框架Netflix的多个开源组件进行了封装，同时又实现了和云端平台以及和Spring Boot开发框架的集成。 Spring Cloud为微服务架构开发涉及的配置管理，服务治理，熔断机制，智能路由，微代理，控制总线，一次性token，全局一致性锁，leader选举，分布式session，集群状态管理等操作提供了一种简单的开发方式[10]。 Spring Cloud 为开发者提供了快速构建分布式系统的工具，开发者可以快速的启动服务或构建应用、同时能够快速和云平台资源进行对接。

第三章 软件系统设计

3.1 项目结构设计

秀视频平台采用MVC三层架构模式，通过springboot框架整合springmvc，mybatis进行搭建。其项目结构图如图3-1所示：

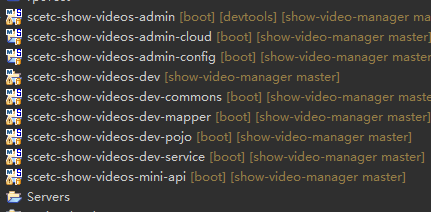


图3-1项目结构图

其中各个模块的介绍如下:

(1).scetc-show-videos-admin 后台管理系统，主要是对小程序端进行管理。

(2).scetc-show-videos-cloud 分布式注册中心，主要是

(3).scetc-show-videos-config 分布式配置中心

(4).scetc-show-videos-dev 小程序端后台，其中scetc-show-videos-dev为一个聚合工程，其分为以下板块,scetc-show-videos-dev-mapper 数据访问层，scetc-show-videos-dev-pojo 实体层，scetc-show-videos-dev-service 服务层，scetc-show-videos- commons 相关的工具类，scetc-show-videos-mini-api 接口层

其中scetc-show-videos-admin后台管理系统详细设计如图3-2所示：

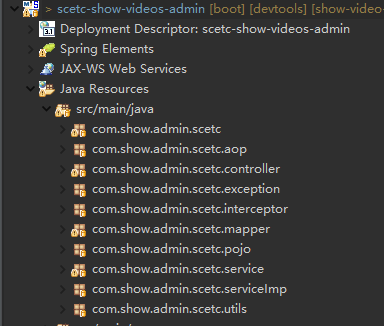


图3-2 scetc-show-videos-admin详细结构

其中com.show.scetc.aop主要实现了对日志的记录，采用aop环绕切面编程的方式，在功能触发之前记录日志。com.show.scetc.controller 是三层架构中的控制层，com.show.scetc.exception统一异常处理，主要的作用是将异常统一处理，由于项目中默认的异常页面是非常不友好，所以封装了一层统一异常处理类对异常进行统一处理。com.show.scetc.mapper是数据接口层，主要是对数据库进行crud操作，pojo层是实体层，在项目中存放相关的实体类，com.show.scetc.utils.工具类，保存相关的工具类比如日期转换，空字符判断等。com.show.scetc.interceptor 拦截器，主要是对权限进行拦截，防止部分功能被越权访问，com.show.ervice包中放置业务层接口，用于规范功能模块的实现方法；com.show.serviceImpl包中放置业务层实现类。

3.2 界面设计

3.2.1小程序端界面设计

视频详情运行效果图如图3-3所示

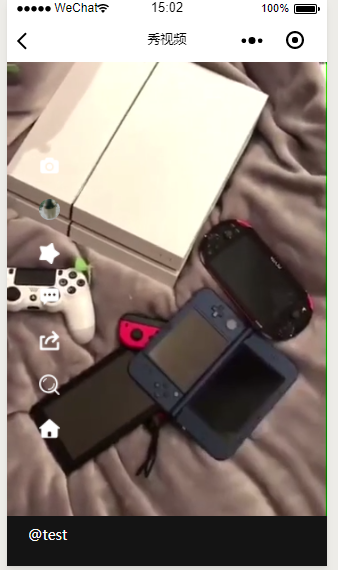


图3-3视频详情页面

在该界面中实现了视频的点赞，分享，评论、搜索、上传、返回首页功能。

首页如图3-4所示：



图3-4程序首页

在程序的首页中，我们可以看到三大分类，其中分别为美拍类、美食类、美妆类，用户可以将视频发布到不同的专栏下，也可以浏览不同专栏的视频。

3.2.2 后台管理系统界面设计

主页如图3-5所示

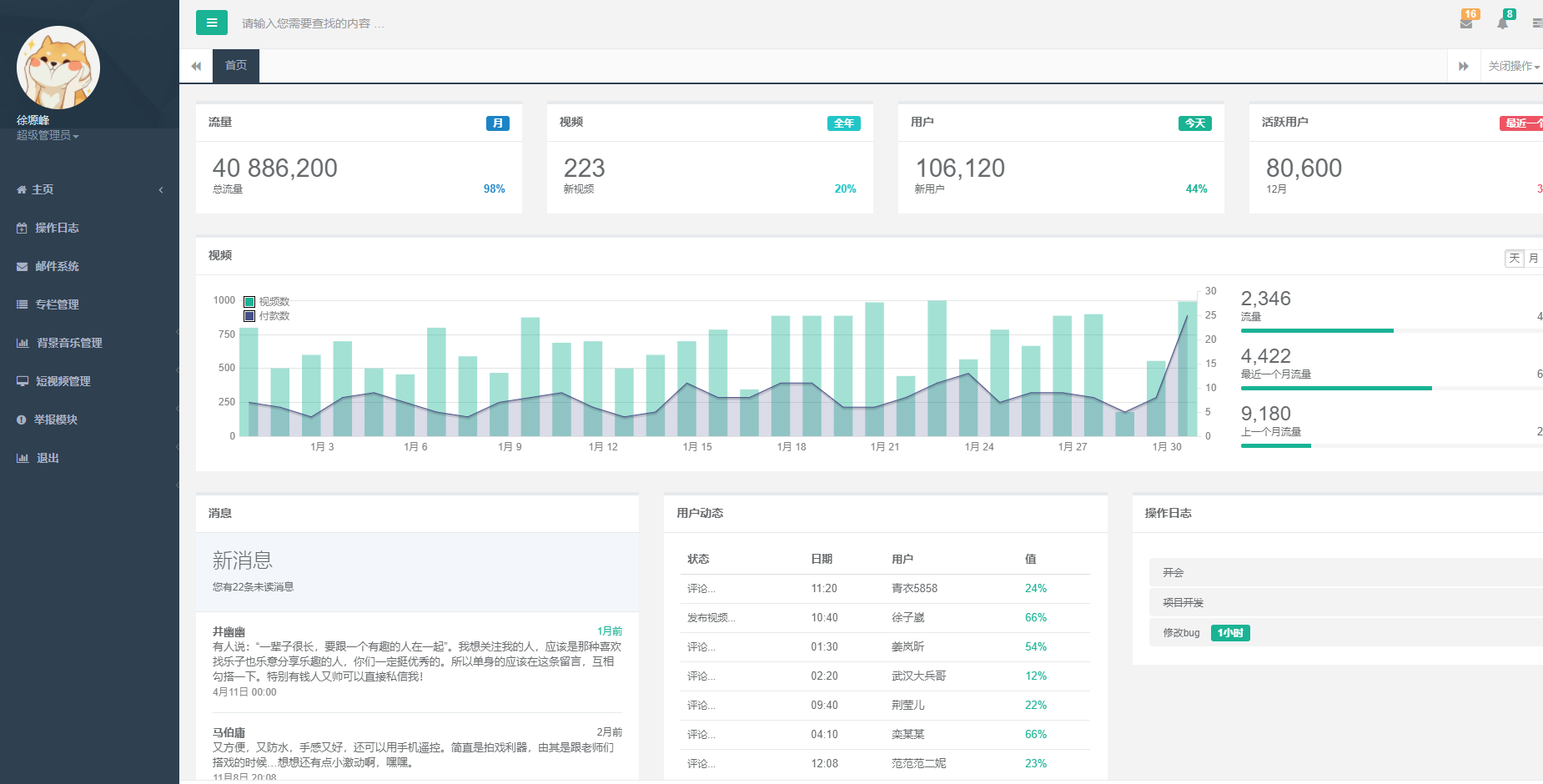


图3-5主页

在后台管理系统主页中，我们将视频数量，用户数量以可视化的方式展现给管理员，并在右下方显示操作日志，用户动态等。

视频管理界面如图3-6所示：

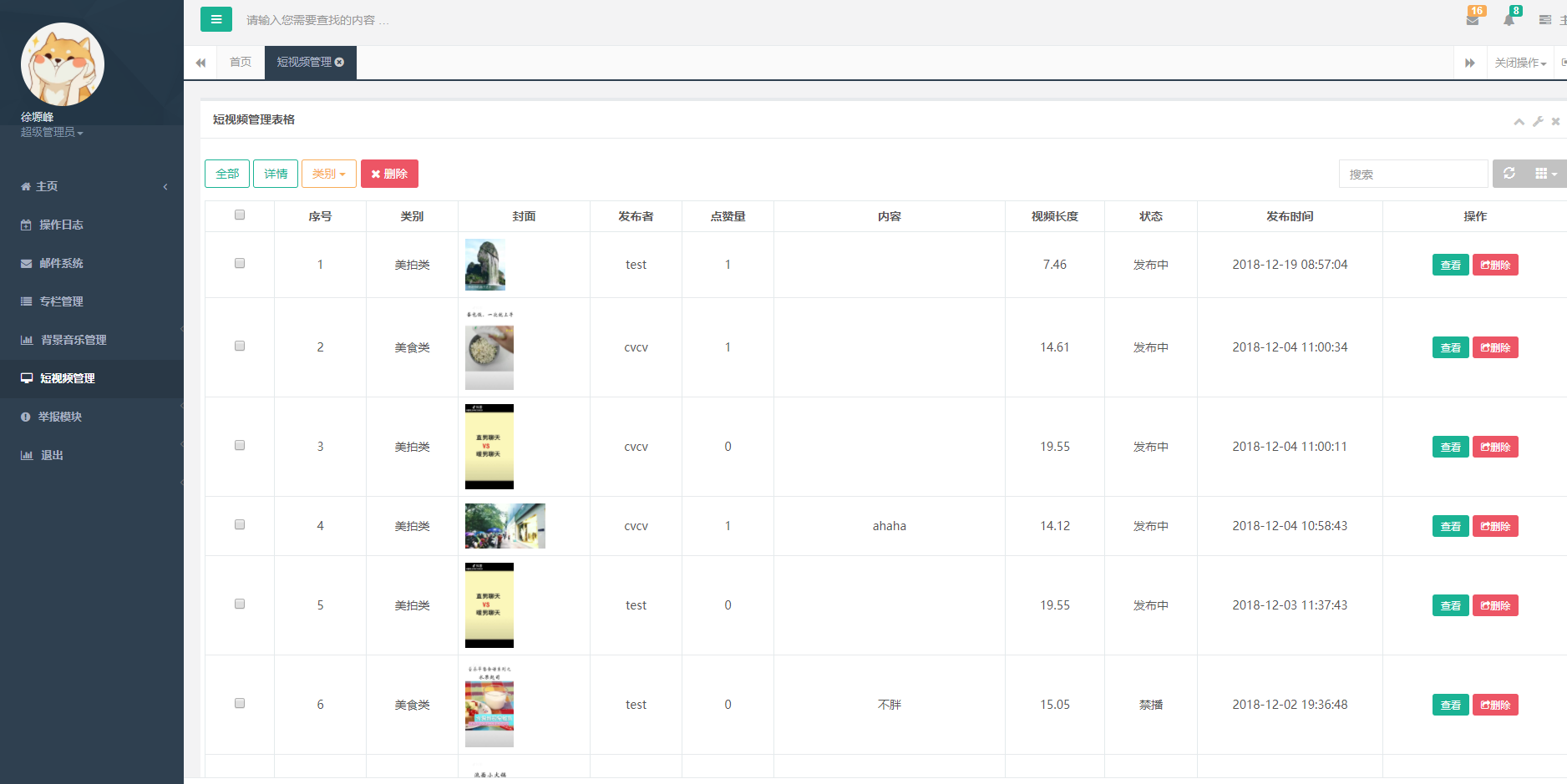


图3-6视频管理界面

在视频管理界面中，我们可以对短视频进行查看，删除、查询。

3.3 功能层设计

在软件应用中，表示层的作用是和用户交流。用户可以通过表示层实现自己想要的功能，而真正为用户实现功能的不是表示层而是功能层。功能层起到的作用是连接界面表示层和数据层，作为连接的桥梁，功能层无疑是整个系统中体现核心价值的好坏。功能层在设计上应该遵循高内聚、低耦合的原则，即模块内部的元素之间紧密关联，同时模块之间的关联程度不大，这样可以有效地提高系统的可维护性和可扩展性。本文将着重讲解关键的几个功能层以及核心功能的代码设计。

3.3.1 小程序端功能设计

小程序登陆功能设计，其中小程序登陆流程图如图3-7所示：

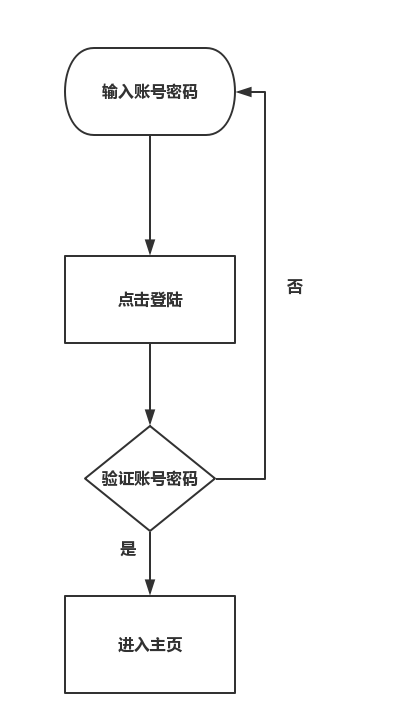


图3-7小程序登陆流程图

其中用户登陆的流程如下：用户在login登陆页面中输入账号密码，点击提交，这时通过微信小程序的wx.request请求将账号密码以参数的形式访问后端相应的接口，在后端对账号密码进行判断，如果账号密码与数据库中保存的账号密码一致，则返回状态码200。小程序端通过success:function(res)取得返回结果，通过结果中的status（状态码）来判断是否登陆成功，如果登陆成功，则跳转到相应的个人页面

小程序的视频上传设计流程图如图3-8所示

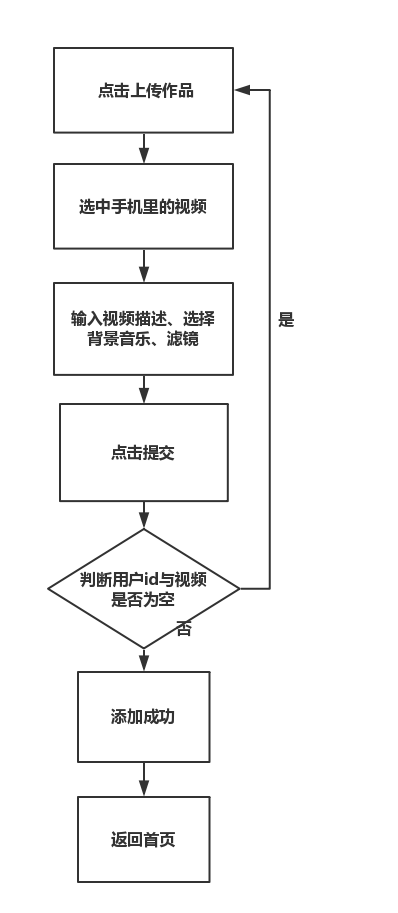


图3-8视频上传流程设计

如图3-6所示，用户发布视频的主要环节有：点击上传进入chooseBgm页面，在该页面中选择背景音乐，滤镜，填写视频描述，点击上传upload获取视频对象的属性以及视频文件，并以request请求的方式访问服务器，在后端，将视频保存在本地，并保存视频相关信息到数据库中，最后返回视频上传成功的状态码200，小程序端通过状态码来判断视频上传成功，视频上传成功后返回主页。

3.2.2 小程序端代码实现

其中前台登陆关键代码如下:

wx.request({

url: serverUrl + '/login',

method: "POST",

data: {

username: username,

password: password

},

header:

{

'content-type': 'application/json' },

//回调函数

//返回

success: function (res){…}

其中wx.request为微信官方api封装的方法，在method中可以设置以post请求访问，可以携带参数data访问后台服务器，并通过success回调函数对页面进行渲染。

**后台服务器登陆代码设计如下：**

@ApiOperation(value = "登陆", notes = "用户注册的接口")

@PostMapping("/login")

public XyfJsonResult login(@RequestBody Users user) throws Exception {

if (StringUtils.isBlank(user.getUsername()) || StringUtils.isBlank(user.getPassword())) {

return XyfJsonResult.errorMsg("小主,账号密码不能为空哦");

}

if (userService.queryUsernameIsExist(user.getUsername())) {

Users userSource = userService.getUser(user.getUsername());

if(userSource.getPassword().equalsIgnoreCase(MD5Utils.getMD5Str(user.getPassword()))) {

user.setPassword("");

UsersVo userVo = setUserRedisSessionToken(userSource);

return XyfJsonResult.ok(userVo);

} else {

return XyfJsonResult.errorMsg("小主,你的账号密码错误");

}

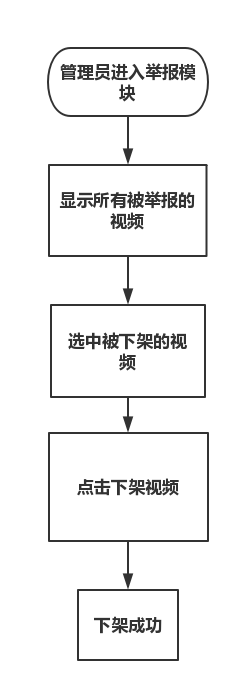
} else {

return XyfJsonResult.errorMsg("小主,你的账号密码错误");

其中获取到前端提交的数据，并通过用户名查询数据库中是否存在这样的用户，如果存在则将用户信息保存到redis缓存数据库中，并响应状态码200 表示登陆成功。

3.3.3 后台管理系统功能设计

其中后台管理管理系统举报模块中下架视频设计流程图如图3-9所示：



3-9视频下架流程图

如图3-7所示:管理员下架视频的主要流程有:进入举报模块，查看所有被举报的视频，选中被举报的视频，通过js获取其id，并通过ajax请求提交给后台服务器，将指定的视频的status状态码更新为2，前台小程序的视频查询的sql语句中有status =1 这样被下架的视频就不会出现在前台小程序中了

3.3.4 后台管理系统代码实现

$.ajax({

url : '../../videos/updateVideos',

type : 'post',

data : param,

dataType : 'json',

success : function(data) {

if (data.status == 200) {

$("#allBgm").bootstrapTable('refresh');

swal("更新成功", "", "success");

} else {

swal("更新失败", "请重新操作", "error");

}

error : function() {

swal("更新错误", "请重新操作", "error");

通过ajax请求访问服务器的接口，并携带参数。在服务器端通过@RestController注解来支持，返回json对象。

**后台管理系统服务器关键代码如下：**

@RequestMapping("/updateVideo")

public XyfJsonResult updateVideo(String status, String id) {

if (status.equals(DELETE)) {

videoService.delete(id);

} else if (status.equals(UPDATE)) {

videoService.update(id);

}

return XyfJsonResult.ok();

}

其中通过传入的status状态码以及id来判断并执行删除操作。最后返回状态码200.

3.4 数据库设计

3.4.1 数据模型设计

这里主要是对Mysql数据库的讲解，关于Redis数据库的设计将会在本章最后章节给出。依据需求和功能模块的设计，总结了系统在数据层面上涉及到的实体以及实体之间的关系，其中包含管理员、用户、视频、背景音乐的关联关系得出如图3-10所示的系统实体关系图：

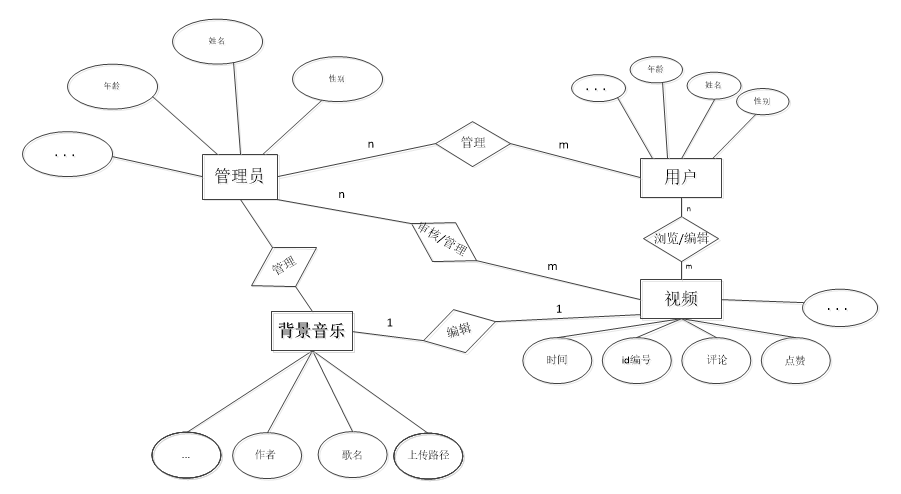


图3-10系统实体关系图

秀视频平台的数据模型图可以在需求分析上建立，根据分析报告中系统需要实现的功能可以确定系统中所需要的实体类有十一个，管理员表，用户表，视频表， \_举报内容表，背景音乐表，短视频表，配置表，专栏表，user\_用户粉丝表关系表， \_用户与视频关系表，搜索记录表, 采用数据库建模软件MySQL Workbench 6.3 CE 对数据库进行设计，其中项目中数据模型设计图如图3-11所示

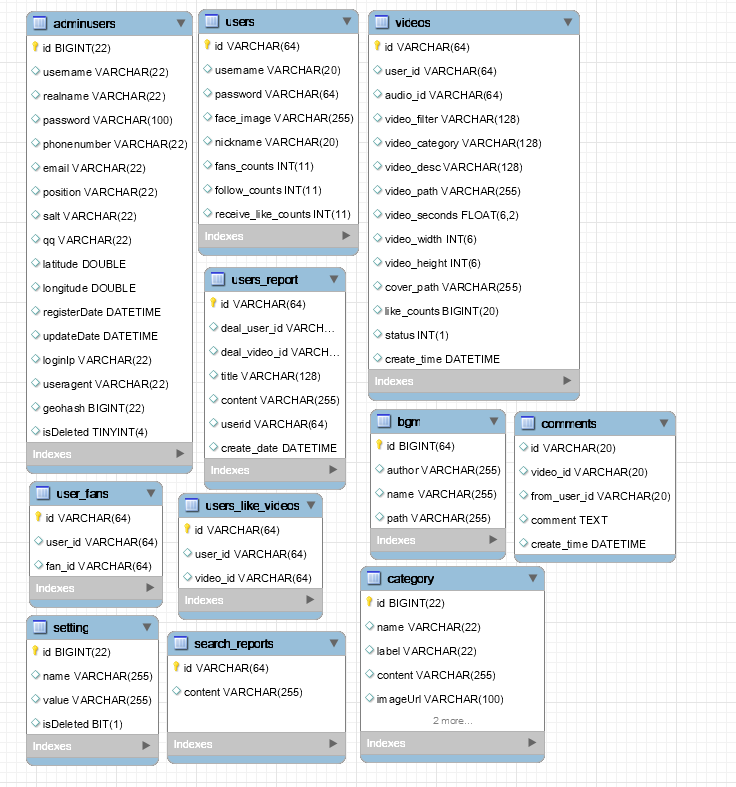


图3-11数据模型设计图

3.4.2 数据层设计

数据层是以功能层为基础来进行设计的，是存储系统中所有数据的地方，数据都是对应的功能层的各种操作以及功能层对数据库的连接。

本系统中核心数据表及其关键字段列举如下：

(1)其中users用户表，其中user表包含了用户的基本信息，编号，姓名，账号，密码，角色的粉丝数量，角色的关注数量，角色的被赞数量 。如表4-1所示:

表4-1 用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 可否为空 | 数据类型 | 备注 |
| id | no | varchar（64） | 主键 |
| username | yes | varchar（20） | 用户的名称 |
| password | yes | varchar（64） | 用户密码 |
| face\_image | yes | varchar (255) | 头像保存路径 |
| nickname | yes | varchar（20） | 用户的昵称 |
| face\_counts | yes | int（11） | 粉丝数量 |
| follow\_counts | yes | int（11） | 关注数量 |
| receive\_like\_counts | yes | int（11） | 获赞的数量 |

（2）其中videos视频表，其包含了视频的基本信息，视频的长度、宽度、视频的描述、视频的分类名称、视频的存放路径、视频的被赞数、视频的状态等，如表4-2所示：

表4-2 视频表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 可否为空 | 数据类型 | 备注 |
| id | no | varchar（64） | 主键 |
| user\_id | yes | varchar（20） | 上传用户id |
| audio\_id | yes | varchar（64） | 背景音乐的id |
| video\_filter | yes | varchar (255) | 视频采用的滤镜 |
| video\_category | yes | varchar（20） | 视频的分类名称 |
| video\_desc | yes | int（11） | 视频的描述 |
| video\_path | yes | int（11） | 视频的存放路径 |
| video\_seconds | yes | int（11） | 视频的长度 |
| video\_width | yes | int（11） | 视频的宽度 |
| video\_height | yes | int（11） | 视频的高度 |
| cover\_path | yes | int（11） | 视频的图片路径 |
| like\_counts | yes | bigint(20) | 被赞的数量 |
| status | yes | int(1) | 状态 |
| create\_time | yes | datetime | 创建时间 |

（3）其中comments评论表，其中主要包含了用户的编号，视频的的编号，如表4-3所示：

表4-3 评论表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 可否为空 | 数据类型 | 备注 |
| id | yes | varchar（20） | 主键 |
| video\_id | yes | varchar（20） | 视频的id |
| from\_user\_id | yes | varchar（20） | 用户的id |
| comment | yes | text | 评论内容 |
| create\_time | yes | datetime | 创建时间 |

（4）其中search搜索结果表，该表主要是用来保存用户的搜索关键字，用户的编号，这些关键字将会在未来被用于大数据推送。如表4-4所示：

表4-4 搜索结果表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 可否为空 | 数据类型 | 备注 |
| id | no | varchar（64） | 主键 |
| content | yes | varchar（255） | 关键字 |

3.4.3 其他数据库设计

由于项目中不仅使用了Mysql数据库，考虑到项目中对数据库的操作负担较大，我们也采用了Redis缓存数据库进行辅助，目的是为了降低Mysql数据库的压力。其中Redis数据库主要使用了key-value的存储方式，以及List存储方式。项目中主要职责是将用户的登陆信息，视频的查询信息，操作日志存放到redis数据库中，这样做的目的是因为redis数据库中的数据大都存放在内存中，从内存中读写数据的速度将会是从硬盘中获取数据的几十倍。提高了软件的运行效率，降低了mysql主数据库的压力。

第四章 系统分析与调试

4.1测试原理

测试是软件开发过程中不可或缺的过程。对于已经实现系统基本功能的项目，必须对其功能进行测试。在验证系统功能正确、修改已存在错误之后才可以发布，并且在后期要进行长期的维护。对系统进行测试可以检测系统的各功能模块，对未达到预期要求的功能进行完善，之后反复对其进行迭代测试，直至测试所得数据能够完成系统预期所需各项功能和用户需求为止。现实中，人无完人，技术产品亦是如此，但对于技术类的产品来说有些错误如果不能被及时发现并改正，有可能就会造成系统的崩溃，同时给使用系统的用户或企业带来不必要的损失。因此，进行系统测试是保证系统稳定性、功能正确性非常重要的过程。我们主要对系统进行以下四个方向测试。

4.1.1 单元测试

对软件中的最小可测试单元进行检查和验证。对于单元测试中单元的含义，一般来说，要根据实际情况去判定其具体含义，如Java里单元指一个类。

4.1.2 功能测试

[功能测试](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD%E6%B5%8B%E8%AF%95/10921202)就是对产品的各功能进行验证，根据功能[测试用例](https://baike.baidu.com/item/%E6%B5%8B%E8%AF%95%E7%94%A8%E4%BE%8B)，逐项测试，检查产品是否达到用户要求的功能。

4.1.3 接口测试

接口测试就是对后端提供的每一个接口进行测试，确保前端页面通过ajax请求能够正确的获取到数据。

4.2测试环境

进行单元测试、集成测试、功能测试。所需的环境如下：

eclise,postman,swagger2.

4.3单元测试用例

在程序中通过@Runwith ，@SpringBootTest注解来进行单元测试，单元测试用例相关代码如图5-1所示：

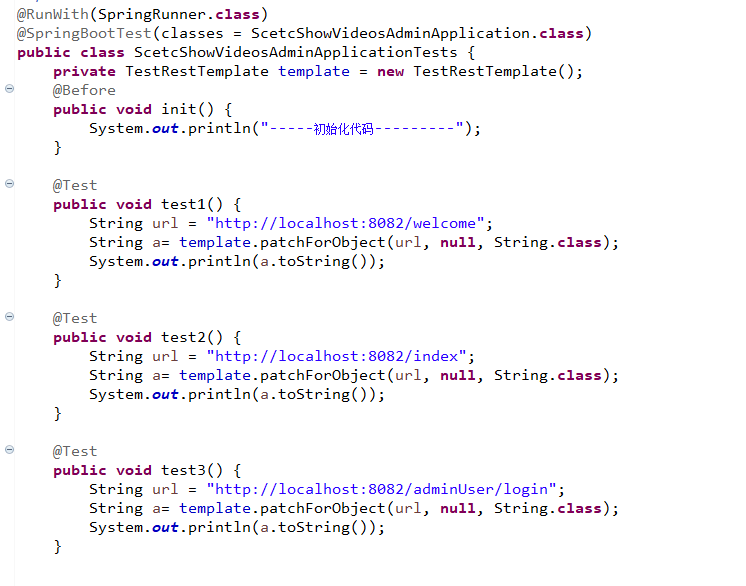


图5-1单元测试用例

通过结果可表明，该系统功能可以正常使用。其他测试都和此案例相同，通过http请求访问api返回数据，

4.4功能测试用例

能完整性测试是系统测试最常见的一种形式，采用功能测试用例来检查系统各项功能的正确性和完整性。系统功能正确性表现在功能是否能够正常使用，完整性表现在功能是否满足用户需求。测试用例针对系统每个功能模块进行，对每个功能点分别测试。

4.4.1 短视频管理测试用例

如表5-2所示：

表5-2 短视频管理测试用例

|  |  |
| --- | --- |
| **功能模块** | 短视频管理 |
| **用例描述** | 进入短视频管理模块，对短视频进行搜索、编辑、删除等操作，然后查询视频信息列表进行验证 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能描述** | **输入条件** | **操作步骤** | **预期结果** | **是否通过** |
| 查询视频信息 | 用户名称、视频内容 | 1.点击视频管理进入视频列表  2.输入视频名称点击查询按钮 | 符合查询条件的视频信息列表 | 是 |
| 修改视频信息 | 视频id | 1点击视频管理进入视频列表  2.选中视频或者输入视频id点击编辑 | 1.进入视频编辑页面  2.提交编辑内容并提交修改成功 | 是 |
| 删除视频 | 视频id | 1点击视频管理进入视频列表  2.选中 | 1.将视频从列表中删除 | 是 |

4.4.2 后台音乐管理测试用例

如表5-3所示：

表5-3 音乐管理测试用例

|  |  |
| --- | --- |
| **功能模块** | 背景音乐管理 |
| **用例描述** | 进入背景音乐管理模块，对音乐进行搜索、编辑、删除等操作，然后查询音乐信息列表进行验证 |

（续上表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能描述** | **输入条件** | **操作步骤** | **预期结果** | **是否通过** |
| 查询音乐信息 | 作者、音乐名称 | 1.点击视频管理进入音乐列表  2.输入音乐名称点击查询按钮 | 符合查询条件的音乐信息列表 | 是 |
| 修改音乐信息 | 音乐id | 1点击音乐管理进入视频列表  2.选中音乐或者输入音乐id点击编辑 | 1.进入音乐编辑页面  2.提交编辑内容并提交修改成功 | 是 |
| 删除音乐 | 音乐id | 1点击音乐管理进入视频列表  2.选中 | 1.将音乐从列表中删除 | 是 |

4.4.3 用户登陆注册测试用例

如表5-4所示：

表5-4 小程序用户登陆注册测试

|  |  |
| --- | --- |
| **功能模块** | **登陆测试模块** |
| **用例描述** | 进入微信小程序，根据已有的账号进行登陆注册测试 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能描述** | **输入条件** | **操作步骤** | **预期结果** | **是否通过** |
| 登陆测试 | 账号、密码 | 1.进入小程序端2输入账号密码3.点击登陆 | 1.登陆成功 | 是 |
| 注册测试 | 账号密码 | 1.进入小程序端注册页面2.输入账号密码3.点击注册 | 1.注册成功  2.返回登陆成功 | 是 |

4.5接口测试

项目中我们采用了swagger2文档测试，方便对每一个接口进行测试，其测试的方法通过@ApiOperation(value = "登陆", notes = "用户注册的接口")注解来支持。其中swagger2测试界面如图5-5所示：

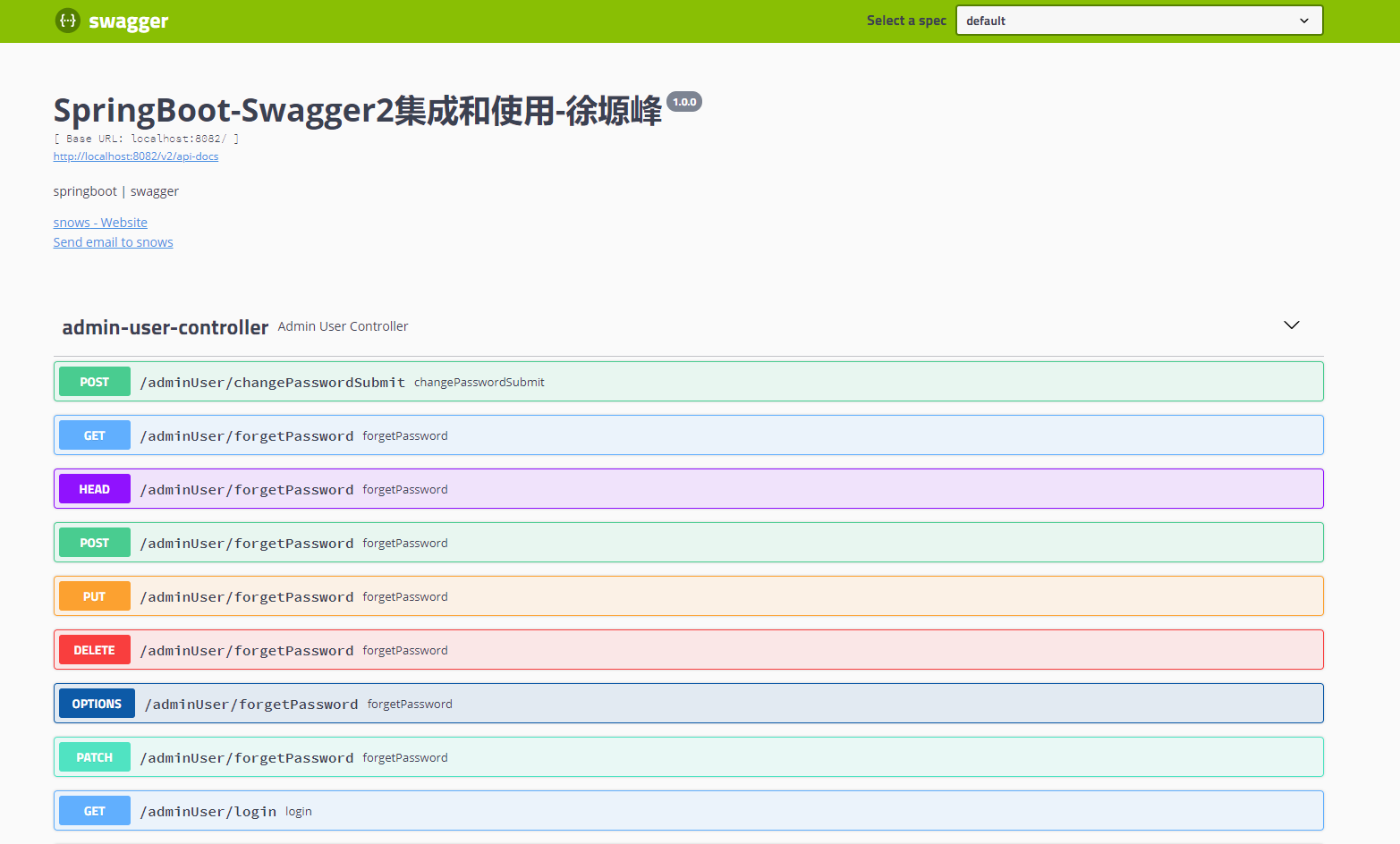


图5-5 swagger2测试界面

通过swagger2可视化测试可以很方便的对每一个接口进行测试

4.6测试结论

测试用例针对系统每个功能模块进行，对每个功能点分别测试，测试结果表明，该系统各模块功能可以正常使用，与预期的结果相符，满足了系统的相应功能需求。

第五章 软件安装

5.1安装准备

系统运行的软件环境需要操作系统、数据库管理系统、网络通讯协议、集成开发环境和系统测试工具等。

5.1.1.操作系统环境

（1）服务器环境支持Windows 2003 Server、Windows 2008 Server、Linux（CentOS 6.6）等网络服务器操作系统。

（2）服务器需安装JRE（Java运行环境），Tomcat8.0服务器

（3）网络数据库采用Mysql5.6版本

5.1.2.集成开发环境

（1）系统开发工具使用Eclipse STS4版本。

（2）开发框架使用SpringBoot 1.58版本。

（3）微信小程序开发工具

5.1.3.网络环境

基于TCP/IP协议内部局域网

5.2安装流程

5.2.1 配置

项目中使用了mysql数据库以及redis数据库。故需要现在配置文件中配置账号密码等相关信息，配置文件存放在每一个项目application.properties文件中。

5.2.2 分布式组件安装

由于该项目所有微服务都依赖于分布式配置中心，所以应该优先安装分布式组件，首先在控制台运行 java –jar scetc-show-videos-cloud 安装分布式注册中心，其次运行java –jar scetc-show-videos-config 将分布式配置中心安装。

5.2.3 小程序安装

小程序前台安装：在微信小程序开发工具中将前台代码scetc-show-videos--pages上传到微信小程序服务器，前台安装完毕。

小程序后台安装：将项目中的war包放入tomcat服务器下的webapp目录下并在bin目录运行tomcat服务器。小程序后台安装就完成了。

5.2.4 后台管理系统安装

后台管理系统依然采用的jar包部署，运行方式如分布式组件一样，在控制台中运行java –jar 安装scetc-show-admin后台管理系统。

结论

当今最火爆的短视频项目，最流行的平台微信平台，最新的技术理念微服务架构，创作出我们的秀视频平台，总的来说，此次系统开发大体达到了预期的目标，实现了需求中的功能需求，但是还有很多可以扩展的地方，比如分布式理念体现还不够完善，可以将项目中的邮件功能，短信功能分离出来，形成单独的微服务，其次在不同服务器中部署软件，就一定要考虑前台小程序与后台服务器中的数据一致性的问题，另外由于服务器数量不够，无法完整的体现出分布式系统的魅力之处。如果有更多的时间，下一步我将对系统进行更深层次的完善，首先，我会考虑使用负载均衡服务器作为流量的入口，通过轮询算法分发到不同tomcat的服务器中。在后台管理系统开发上，使用shiro 安全认证框架对每一个管理员进行权限认证，在部署上，使用docker进行服务部署，由于docker容器与springcloud是天然耦合的，使用docker进行软件的部署可以很大的节约部署的时间，提高软件开发的效率。

致谢

在四川工程职业技术学院三年的大学学习生涯随着毕业答辩的结束也即将落下帷幕，这三年是我人生中最值得回忆的美好时光。在这里，我深深地感受到了工院老师及学生的风采，在这里，我认识了许多辛勤的园丁，结交了一群志同道合的朋友，这一段求学旅程是我人生中宝贵的财富，在这里，我认识到我的女朋友，因为它的鼓励以及支持，让我坚持自己的道路。

首先，我要感谢我的指导老师曹冰老师，从论文的选题到后面的一次次修改，曹老师都耐心指导，她严谨的教学态度、渊博的知识以及谦和的处事方式都深深感染着我。其次，感谢我的同学、学长、朋友，在我写论文遇到困难的时候给予我帮助和鼓励，让我得以顺利完成毕业设计和毕业论文。最后，感谢大学三年遇见的所有老师和同学，谢谢你们来到我的生命里。

毕业在即，祝愿所有人一切顺利！

参考文献

[1] [吴佳妮](http://lib.cqvip.com/ZK/search.aspx?Query=%e5%90%b4%e4%bd%b3%e5%a6%ae&Type=A)，音乐社交短视频软件何以走红以抖音App为例， 2016:1-2

[2] 黄静旋,短视频时代来了，2016:1-2. 2016年，第0卷第2期 112-113页,共3页。

[3] [田志友](http://lib.cqvip.com/ZK/search.aspx?Query=%e7%94%b0%e5%bf%97%e5%8f%8b&Type=A);[周元敏](http://lib.cqvip.com/ZK/search.aspx?Query=%e5%91%a8%e5%85%83%e6%95%8f&Type=A),[田雨](http://lib.cqvip.com/ZK/search.aspx?Query=%e7%94%b0%e9%9b%a8&Type=A) ,微信小程序的媒体价值.2018年第4卷第1期 47-49页,共3页

[4] 张峰, [微服务技术构建大规模web系统的研究](http://www.cnki.com.cn/Article/CJFDTotal-CXYY201722026.htm) 2017年

[5] [彭永勇](http://lib.cqvip.com/ZK/search.aspx?Query=%e5%bd%ad%e6%b0%b8%e5%8b%87&Type=A),[王先兵](http://lib.cqvip.com/ZK/search.aspx?Query=%e7%8e%8b%e5%85%88%e5%85%b5&Type=A)基于互联网应用模式的企业级微服务技术研究《China Computer & Communication》,2018年第0卷第2期 9-11页,共3页.

[6] [胡鹏](http://lib.cqvip.com/ZK/search.aspx?Query=%e8%83%a1%e9%b9%8f&Type=A);[高永平](http://lib.cqvip.com/ZK/search.aspx?Query=%e9%ab%98%e6%b0%b8%e5%b9%b3&Type=A)基于SSM架构的汽车客户服务平台的设计与实现 2017年第13卷第4期 50-52页,共4页

[7] [张雷](http://lib.cqvip.com/ZK/search.aspx?Query=%e5%bc%a0%e9%9b%b7&Type=A) ;[王悦](http://lib.cqvip.com/ZK/search.aspx?Query=%e7%8e%8b%e6%82%a6&Type=A) 基于SpringBoot微服务架构下的MVC模型研究2018年第17卷第4期 1-9页,共9页

[8] [蒙杰](http://lib.cqvip.com/ZK/search.aspx?Query=%e8%92%99%e6%9d%b0&Type=A),[杨生举](http://lib.cqvip.com/ZK/search.aspx?Query=%e6%9d%a8%e7%94%9f%e4%b8%be&Type=A),[施韶亭](http://lib.cqvip.com/ZK/search.aspx?Query=%e6%96%bd%e9%9f%b6%e4%ba%ad&Type=A),  基于SSM轻量级复合框架的科技创新券管理系统研究与实现《[兰州文理学院学报：自然科学版](http://lib.cqvip.com/qk/91153B/)》 2018年第32卷第2期 67-70页,共5页

[9] [李莎](http://lib.cqvip.com/ZK/search.aspx?Query=%e6%9d%8e%e8%8e%8e&Type=A) ,[林珠](http://lib.cqvip.com/ZK/search.aspx?Query=%e6%9e%97%e7%8f%a0&Type=A) ,[郑伟鸿](http://lib.cqvip.com/ZK/search.aspx?Query=%e9%83%91%e4%bc%9f%e9%b8%bf&Type=A) 基于SOA与restful架构的广东省实验室体系管理平台设计《[自动化与信息工程](http://lib.cqvip.com/qk/90639B/)》 2018年第39卷第2期 32-36页,共5页

[10] [周永圣](http://lib.cqvip.com/ZK/search.aspx?Query=%e5%91%a8%e6%b0%b8%e5%9c%a3&Type=A) ,[侯峰裕](http://lib.cqvip.com/ZK/search.aspx?Query=%e4%be%af%e5%b3%b0%e8%a3%95&Type=A) ,[孙雯](http://lib.cqvip.com/ZK/search.aspx?Query=%e5%ad%99%e9%9b%af&Type=A) ,[杨磊](http://lib.cqvip.com/ZK/search.aspx?Query=%e6%9d%a8%e7%a3%8a&Type=A) ;[张小贝](http://lib.cqvip.com/ZK/search.aspx?Query=%e5%bc%a0%e5%b0%8f%e8%b4%9d&Type=A)基于SpringCloud微服务架构的进销存管理系统的设计与实现,《[工业控制计算机](http://lib.cqvip.com/qk/92874X/)》 2018年第31卷第11期 129-130页,共3页