|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | |
| 50da81cb39dbb6fdf8b8fff80b24ab18962b3708 | | |  | | | | ○ **A** **基础理论**  ● **B** **应用研究**  ○ **C** **调查报告**  ○ **D** **其他** | | | | | |
|  | | **岭南师范学院** | | | | | |  | | | | |
|  | **本科生毕业论文（设计）** | | | | | | | | | |  | |
|  | | | | | | | | | | | | |
|  | **儿童疫苗接种管理系统的设计与实现** | | | | | | | | | | |  |
|  | | | | | | | | | | | | |
|  | | | | **二级学院** | **：** | **信息工程学院** | | | |  | | |
| **专 业** | **：** | **软件工程** | | | |
| **年 级** | **：** | **2017级** | | | |
| **学 号** | **：** | **\*\*\*** | | | |
| **作者姓名** | **：** | **\*\*\*** | | | |
| **企业导师** | **：** | **\*\*\* 职称** | | | |
|  | | | | **学院导师** | **：** | **\*\*\* 职称** | | | |  | | |
|  | | | | | | | | | | | | |
|  | | | | **完成日期** | **：** | **年 月 日** | | |  | | | |

**儿童疫苗接种管理系统的设计与实现**

专业名称：软件工程

作者姓名：\*\*\*

企业导师：\*\*\*

指导教师：\*\*\*

**论文答辩小组**

**组长**：

**成员**：

**论文成绩：**

**目 录**

1 引言 1

1.1管理信息系统的定义和发展 1

1.2疫苗接种管理的现状和存在的问题 1

1.3疫苗接种管理信息系统研究的意义 2

2系统开发工具及技术介绍 3

2.1开发工具介绍 3

2.1.1开发工具IDEA概述 3

2.1.2数据库概述 3

2.1.3画图工具概述 3

2.1.4项目管理工具概述 3

2.2开发技术 3

2.2.1后端开发技术 3

2.2.2前端开发技术 3

3系统需求分析 3

3.1 系统定位 3

3.1.1系统作用 3

3.1.2系统架构 4

3.1.3系统角色说明 4

3.2 系统的功能需求 4

3.2.1系统功能说明 4

3.2.2系统用例图 5

4系统概要设计 5

4.1系统功能结构图 5

4.2系统的数据库设计 5

4.3系统流程图 6

5 系统详细设计与实现 6

5.1系统配置 6

5.1.1后端框架搭建 6

5.1.2数据库的配置 7

5.1.3 Redis缓存用户Tokend的实现 8

5.1.4手机短信通知 8

5.1.5导出Excel文件 8

5.2系统目录结构 8

5.3系统功能实现 8

5.3.1疫苗管理功能模块 8

5.3.2预约接种管理功能模块 8

5.3.3儿童管理功能模块 8

5.3.4不良反应管理功能模块 8

5.3.5用户通知管理功能模块 8

5.3.6用户管理功能模块 8

5.3.7系统管理功能模块 8

5.3.8登录注册功能模块 8

5.3.9手机号码修改密码功能 8

5.3.10系统首页 8

6 系统的测试与分析 8

6.1系统测试的目的和原则 8

6.1.1系统测试的目的 8

6.1.2系统测试的原则 9

6.2系统测试阶段 9

6.3对本系统的测试 9

7 总结与展望 9

**基于SpringCloud的慕课后台管理系统的设计与实现**

作者　肖明章　指导教师　李庆华 赵宁

(岭南师范学院信息工程学院，湛江 524048)

**摘　要：**随着互联网的发展，慕课学习已逐渐成为现代获取知识的重要途径，慕课英文缩写为MOOC,意思是大规模开放在线课程。慕课的快速发展也为慕课的管理产生了一系列问题，涉及到视频的播放，除了需要占用过高的带宽，还需要承载较大的并发量，随着用户数据量的增大，慕课的后台管理面临着巨大挑战。传统的单体项目架构在各方面都容易到达瓶颈，功能业务集中在一个项目里，耦合度高，导致后期难以扩展和维护。随着技术的不断更新，微服务架构的出现解决单体架构的痛点，如今微服务已经相当火热。本系统是基于微服务Spring Cloud + Vue的分布式前后分离的慕课后台管理系统，开发本系统目的是为了解决单体慕课后台管理系统项目的各种瓶颈，使慕课视频后台管理更便捷、可监控。

**关键词：**MOOC;慕课后台管理；微服务架构；Spring Cloud;

**Design and Realization of Child Vaccination Manage Information System**

LIN HuZhou

School of Information Engineering, Lingnan Normal University, Zhanjiang, 524048 China

**Abstract:** With the continuous development of vaccination prevention and control in China and the increasing awareness of vaccination, the number of our country vaccinated children is raising, and the drawbacks of the C/S structure vaccination management system of vaccination centers are becoming more and more obvious. At present, the management system based on B/S structure has become a mainstream development mode and has been widely used due to its high flexibility, strong scalability and distributed characteristics. Base on SpringBoot operation framework , SSM development framework to developed a child Vaccination Manage Information system, implementation vaccine management, to make an appointment the vaccine immunization management, children management, adverse reactions management, announcements management, user management, personal information management and system management functions, not only is vaccination center raised the work efficiency and management efficiency, but also reduce the children's vaccination failure rate.

**Key words:** B / S structure Management System; Vaccination; SpringBoot; Distributed

# 1 引言

2020年12月9日清华大学举行了世界慕课大会，这表示慕课的发展前景巨大。随着慕课的发展，课程人数的增多，慕课的管理变得越来越复杂。管理员对课程和人员的管理变得困难。现在，慕课后台管理系统的需求越来越迫切，但是传统的慕课后台管理系统难以解决慕课的高速发展带来的隐患。如日益新增的需求、不断暴涨的数据，这些都使得传统慕课管理系统难以承受。如今微服务技术已经成熟，可以在一定程度上解决传统技术架构的隐患，使用微服务架构技术开发的慕课后台管理系统更具备高可用性、扩展性。

## 1.1慕课的定义和发展

慕课(MOOC)，也叫大型开放性网络课程(Massive Open Online Course)。是现在很火的一种教学模式。MOOC的概念是2008年加拿大学者 Bryan Alexander 和 Dave Cormier 提出的一个概念，它具备大规模、开放性、网络教学等特性。

(一) 大规模性(Massive)

慕课的大规模性，主要涵盖以下几个方面，一是课程数量庞大、课程类型覆盖面广。二是指学习人数规模庞大。与传统的线下教学模式不同，一门课程可能被不同学校的学生、甚至不同群体的人员参加学习，因此用户数量是非常庞大的。

(二) 开放性(Open)

慕课的开放性，无论是什么背景，都可以通过注册账号，免费学习到名校教师、领域专家的课程。慕课的开放性是面向大众的，为热衷于学习的人，提供了一条很好的求学路径。特别对于互联网计算机行业，技术的开放和共享非常重要。而慕课平台就为我们提供了这样一个学习、资源共享的平台。

(三) 网络性(Online)

慕课的发展是源于网络的发展，它是互联网 + 教育的产品，因此具备网络特性。与传统对课程相比，慕课具备网络性使得求学者可以随时随地、不拘束于地点和时间进行学习。并且可选择性也更高了，可以自由选择自己喜欢的类型、教师的课程。由于是视频形式的学习，可重复播放复习，难以理解或者容易忘记的知识点可以通过多看几遍视频，这是一种很好的循环渐进式学习模式。

慕课的发展速度非常快从2008年开始开始出现Coursera、edX、[Udacity](https://baike.baidu.com/item/Udacity" \t "_blank)等大型慕课平台。Coursera作为世界上最大的慕课平台，如今已经拥有超过2500门课程，这些课程覆盖了各门学科，慕课影响巨大，为求学者提供便捷的学习和提升的路径。

在我国，慕课学习人数已到达7000万，慕课学习的主体用户是学生。国内也开始出现很多优秀的慕课平台，像中国大学MOOC、慕课网等。慕课成为了获取知识的天堂。慕课的主体不仅面向学生，而是面向所有求学者，通过慕课学习是非常高效的途径，慕课的意义非凡。

## 1.2慕课后台管理的现状和存在的问题

随着慕课的飞速发展，用户数量的剧增，已经到了千万级别。加上有不同学科的课程新增，课程种类繁多、教师人员变动频繁，这些因素都使得慕课的管理更加困难。在日益增加的业务需求面前，传统的技术架构也面临着各种瓶颈。首先，所有工程代码集中在一个工程了，代码量的不断增多使得单体的慕课项目日益膨胀，变得难以维护的扩展。其次，慕课产生的数据量的逐渐增加也使得原来的系统查询检索、统计变得困难，如果引入新的技术去解决问题又容易对系统其它功能造成影响，这些不管是对用户还是管理员来说都是很不友好的。最后，管理员权限和职责不明确，管理员往往岗位不同所具备的管理权限也是不同的，很多慕课后台管理系统都没有做到分权管理，权限不是动态分配往往导致无法适应频繁的岗位的变更的情况，权限的耦合对系统对管理员的管理非常不方便。

## 1.3基于微服务慕课后台管理系统研究的意义

为了满足“互联网+教育”的需求，解决视频课程、慕课注册人员数量庞大管理困难的问题，开发慕课后台管理系统对视频课程、教师、人员、分离等进行可视化管理。慕课后台管理系统，也方便了教师上传视频课程资源、对课程进行管理和监控，为教师用户和管理员提供一个良好操作平台。

在这个系统中，管理员能系统化地对人员和讲师进行监控和管理，及时对违规人员或者违规课程进行处理。开发本系统的目的是为了弥补现有网课系统的健康管理难等问题。

基于微服务的慕课后台管理系统的意义主要集中以下几点：

（1）简单、可视化管理

通过模块化慕课管理，课程管理、教师管理、分类管理、系统管理等。不管是管理员还是教师，都可以通过页面上清晰明了的数据和操作按钮快速对相关信息进行管理，并且还有图表展示统计的重要数据趋势，目的是为了使网课管理更便捷、更高效。

（2）免费、面向大众

慕课的意义就是为大众提供免费的、多样的、高质量的课程，同样慕课后台管理系统也遵循着这样的原则，面向大众，杜绝收费，老师、学者、某一领域的专家等自由分享知识，让更多人可以在上面收获知识。

（3）技术前沿

随着技术的更新换代，近几年微服务的概念已经非常火，目前很多大中小型公司已经开始尝试使用微服务架构。Spring也推出广受开发者喜爱的Spring Cloud框架，Spring Cloud集成各种微服务常用组件帮助我们快速搭建微服务项目。基于SpringCloud框架的慕课后台管理系统，使得慕课管理系统的架构得以脱胎换骨，重获新生。使用前沿的开发技术，一方面可以吸引更多年轻主力开发，还可以解决老旧技术的耦合度高难以扩展等问题，增加系统的可用性、可维护性，大大减少开发和迭代升级的人力成本。

# 2系统开发工具及技术介绍

## 2.1开发工具介绍

### 2.1.1开发工具概述

IDEA

IDEA是一个集成了各种java开发的集成工具，IntelliJ IDEA具备简洁好看的开发界面，除了有友好的代码提醒、修改历史记录的查看与恢复等功能。还集成了各种环境和工具插件，可以供开发者灵活选择。它的优秀也不仅于此，良好得debug能力也使它比传统java开发工具eclipse更受欢迎, 现在对于IDEA在业界的地位已经非常高，对于java开发者来说，使用IDEA进行开发绝对是首选。

### 2.1.3项目管理工具概述

Git是一个常用的版本控制工具，可以对代码进行合理管理，是团队开发必不可少得工具。支持拉取、提交保存、推送、分枝管理、合并、版本回退等操作。

使用Git作为项目管理和版本控制工具有助于提升工作效率，关联远程仓库使得代码在一台机上更新后可以迅速更新到其他开发人员的本地，灵活使用git是开发人员的必备技能之一。

### 2.1.4数据库概述

本系统使用到的数据库有MySql、Redis、Elasticsearch。以下对这三种数据库进行简单描述。

**MySql:**

MySql是一个优秀的开源关系型数据库，从上世纪开始就开始使用至今，一般来说，其他技术三五年便更新一大版本的适合，MySql仍然以优秀的结构和设计，一直保存至今，被广大公司和开源开发者最为首选的开源数据库之一，并扛住了各式各样的业务模型和数据结构。直到现在，针对于MySql的各种高级操作也频频出现在开源项目中，如MyCat，分库分表，弹性扩容等等。

目前MySql有俩个较大的版本分支，5.0+与8.0+ ，MySql在5.5.5开始就支持InnoDB存储引擎，InnoDB是目前使用最广的存储引擎，它具备回滚、提交等事务特性。本系统使用的是5.7版本，5.7是目前比较稳定的版本。

**Redis:**

与mysql不同的是，Radis是一种Key-Value类型的数据库，是基于内存的数据库，因此redis的读取效率非常高，并且可以抗住每秒十万的并发读写。一般使用redis对读多写少的场景下做缓存，以提高系统的并发承受能力。

**ElasticSearch:**

ElasticSearch是一个优秀的数据检索、数据分析引擎，所有需要对外提供检索或分析的数据都可以存储到ES。ES具备以下几个优势: 1、全文检索 2、多样化的聚合分析，提供良好的图表分析能力。3、节点可横向扩展，具备大数据量的搜索和聚合。

### 2.1.2数据库可视化工具概述

## 2.2开发技术

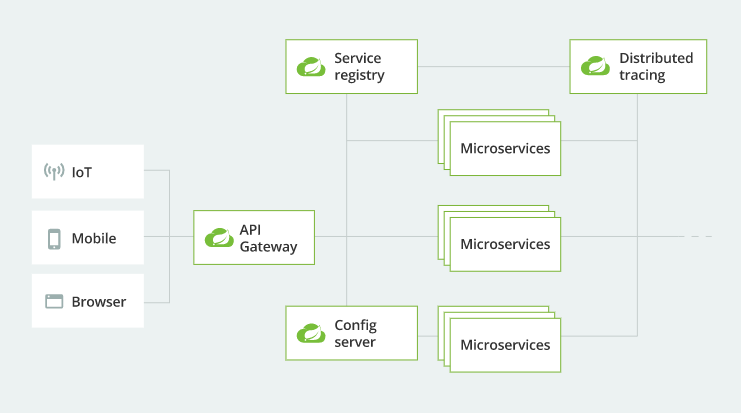
### 2.2.1后端开发技术

微服务

微服务是一种分布式架构，把一个项目工程划分为多个细小的服务。多个细小的服务共同组成一个软件项目。每一个微服务都可以单独运行，有自己的堆栈内存，甚至可以在不同的服务器上跑。随着业务的扩展，单体的痛点逐渐暴露，许多公司的软件产品也渐渐从单体走向微服务化。现在微服务已经成为未来一种趋势。

**Spring Cloud**

后端使用springcloud框架搭建微服务分布式系统，springcloud是spring家族框架之一，它集成了各种微服务组件方便我们搭建微服务架构，其中包括服务注册中心Eureka、Api网关Gateway、服务熔断Hystrix、服务调用Feign、配置中心等一系列微服务常用组件。



**注册中心：**所有微服务需要向注册中心注册，用于记录各个服务的元信息。每个微服务都需要定时向注册中心发送心跳信号，注册中心根据心跳信号来判断服务是否可用。目前注册中心的实现比较多，比较主流的有Google开源的Consul、Apache的Zookeeper、Netflix开源Eureka、阿里的Nacos等。本系统使用Eureka作为注册中心。

**网关服务：**网关是所有的微服务访问入口，所有请求都先经过网关，网关再根据路径路由转发到对应的微服务。在网关层可以做很多事，像权限验证、负载均衡、限流等等。目前比较常用的网关实现有Zuul 和Spring Cloud Gateway。Zuul是Netflix公司的开源。在Spring Cloud中已经不推荐使用了。本系统使用Spring Cloud Gateway作为网关服务。

**服务调度：**

不同微服务之间会存在互相为彼此提供服务的情况，例如订单服务需要向用户服务获取用户信息。他们之间可以通过Feign来实现。Feign是基于Http请求方式的，里面使用面向接口的动态代理，我们只需要写好接口里的方法及注解，就能实现服务间的调用，非常简单方便。

**服务熔断：**

在微服务群体中，若其中一个服务挂了，很容易引起服务“雪崩”的情况，所以需要做熔断处理。使用Hystrix做服务熔断，在一个服务挂掉的情况，迅速做出熔断处理，避免不断请求等待消耗资源导致其他服务也挂掉。

**消息队列:**

RabbitMQ是基于Java实现的消息队列，具有可靠性安全性，消息队列最大的作用是做到生产与消费之间解耦。常用于异步处理和微服务间消息通知。

**SpringB oot：**

springboot是简化了spring繁琐的配置，SpringBoot做到了少配置、快速、简单搭建一个web项目，每一个微服务都是一个SpringBoot应用。可以说SpringCloud是基于SpringBoot上的。

**微服务的优势：**

1. 不同服业务相对独立，务职责分明，解决单体项目过于“臃肿”、“耦合”等问题，易于后续扩展。
2. 资源分配可以更加灵活，给不同服务根据实际情况配置不同的内存资源，进行比较精准细粒度的资源控制。
3. 每个服务粒度小，灵活，便于开发，不同服务可以使用不同技术开发，技术选择灵活。
4. 更容易横向扩展，对负载高的服务进行集群，使其可以达到可以承载更高的负荷。
5. 即使某一个服务挂了，只是某部分功能不可用，其他服务任可正常工作。

### 2.2.2前端开发技术

**Vue.js**

Vue.js是目前比较流行的前端渐进式框架。和其他前端级框架不同的是，Vue具备简单、灵活、高效等特点，相对于其他前端框架来说Vue语法也相对简单，容易学习，通过vue-cli能快速地去搭建一个单页面的前端项目。如今流行前后分离模式，Vue现在在国内的使用几乎占半边天，其生态也逐渐完善，社区比较活跃，至今已经发布至Vue3.0版本。

由于3.0刚发布不久，稳定性无法得到保证，本管理系统使用的是Vue2.0,并且使用Element-UI组件库搭建页面样式，Vue搭配Element-UI是目前的主流开发模式。

**Bootstrap**

Bootstrap是 Twitter公司开源的前端框架，其封装了大量css样式、各种前端通用组件、js插件等等，还有它的栅格系统使前端的布局变得更简单。不仅如此，他还支持响应式布局，也是一个优秀的移动端页面开发辅助框架。他的出现，让即使不懂前端的人也能搭出比较好看的页面，大大降低了前端的开发门槛，前端开发人员使用Bootstrap能快速搭建出符合需求的前端页面。

# 3系统需求分析

## 3.1 系统定位

### 3.1.1系统作用

慕课的用户端，主要是面向广大用户的，为用户提供一个免费并且课程种类繁多，资源丰富的学习平台。

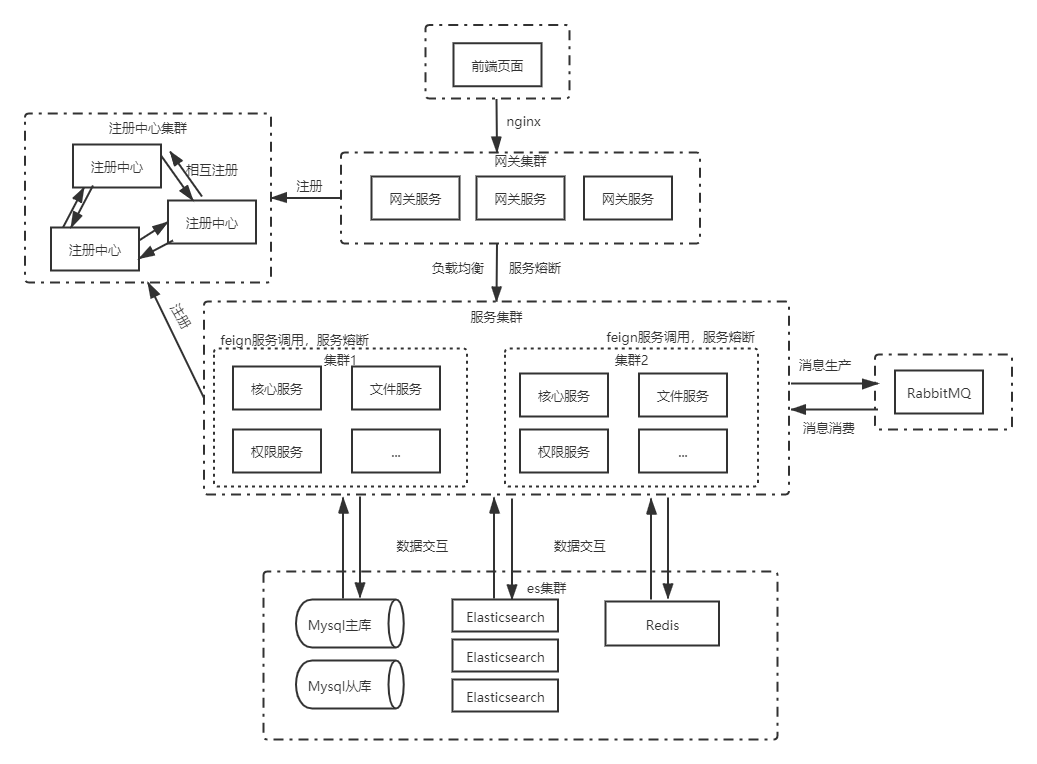
而慕课后台管理系统主要是面向教师和管理员的，主要是对慕课有关的数据进行管理，比如课程管理、章节管理、分类管理、教师管理、用户管理、文件管理等等。开发慕课后台管理系统为了使慕课的管理更加模块化、系统化、简单化。首先是为教师人员开设课程、上传教学视频提供一个路径，二是为管理员监控管理课程和人员提供一个良好的平台。教师可以通过本系统开设课程，管理课程、章节等信息，这使得分享知识变得更简单。管理员通过本系统可以对慕课的教师及课程、分类信息进行审核、修改、删除等操作。除此之外还可以对违规用户进行禁用和踢除下线操作，查看系统的登录日志、统计数据等等。总的来说，慕课后台管理系统可以使得管理页对慕课的管理变得更加高效。

此外，系统使用前沿的微服务架构和各种主流技术组成，打造一个高可用、易扩展的后台管理系统，这使得慕课后台管理系统更具备时代性、技术性。

### 3.1.2系统框架架构

网课后台管理系统，考虑到视频上传和播放可能会占用系统过高的带宽，若使用单体应用则会影响到整个系统的流畅度。本系统决定使用springcloud搭建高可用、可扩展的微服务架构，把视频文件的上传、播放、下载等功能单独划分到一个微服务上。系统由多个微服务组成，分为注册中心、网关服务、核心服务、文件服务、权限服务、报表统计共七个微服务。每个微服务都可以独立运行，使用springboot技术开发，服务间通过Feign调用，并且使用Redis数据库做session共享和“热点”数据缓存，使用RabbitMQ做服务间的通信。在统计报表服务中还集成了Elasticsearch, 让系统具备在千万级别的数据中提供搜索和数据分析能力。

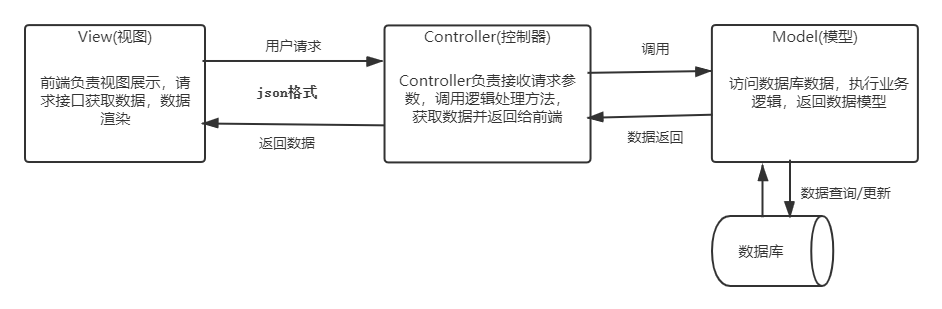
系统架构如图1所示



**图1 系统架构图**

各个组件及微服务以集群方式部署，大大提高系统的并发访问量，使具备系统高可用、可伸缩的能力。为了提高系统的容灾能力，mysql数据库采用主从方式部署，备份使得数据库所在服务器假如发生故障，可从备份数据库中恢复。es集群的部署除了提高容灾能力还可以横向扩展，使得可以存储大数据量，并且保持比较好的数据检索和分析能力。

在整体工程架构上，本工程也采用目前最主流的MVC开发方式，对前后端进行分离。前后端通过ajax技术，使用http协议进行通信，并以轻量级的json数据格式进行数据交换。前端采用的是目前主流的三大前端框架之一的vue进行页面渲染和展示。可以说这一套技术组合是目前比较流行的组合。



MVC是一种典型的软件开发框架，其中包括就是View（视图）， Controller（控制器）， Model（模型）。

MVC模式的最大的优点就是实现视图和业务逻辑的分离，使得本来耦合度高的读取、处理、显示步骤得以解耦，大大提高了业务代码的可重用性。比如我写一个查询的service可以通过Controller被多个视图重用，而不用针对每一个视图都要写对应查询。

SpringMVC是一个比较灵活的一个MVC框架，使得Web开发更加简洁、简单、快速，底层是Servlet，我们不需要关注Servlet如何接收请求和应答，通过简单的几个注解就可以绑定参数、或者返回数据或视图。屏蔽了Servlet细节和web.xml里繁琐的配置，这大大加快了开发效率，并且代码更加简洁了。

前后分离，前后分离模式已经是现在主流开发模式，在一个项目里通常分前端人员和后端人员，他们之间的开发应该是可以同步进行的，前端通过ajax发送http请求获取数据，渲染，编写页面样式。后端负责接收前端传来参数，编写数据处理逻辑，返回对应数据。

前后端之间是完全分离的，包括前端页面跳转、视图选择等都由前端做处理，后端只负责返回数据。在前后分离模式中，前后端往往约定一种数据格式，比如json，共同维护一份接口文档用来实现前接口对接。

关于儿童疫苗接种管理系统的运行，用户只需操作的电脑安装有浏览器，使用浏览器就可以使用该系统，系统页面简约美观，功能操作方便简单，对不熟悉电脑操作的用户友好度高。系统架构如图1所示。

### 3.1.3系统角色说明

本系统选用RBAC数据模型进行权限控制，RBAC（Role-Based Access Control）是基于角色的访问控制。用户并不是直接去访问资源的，用户通过角色去访问资源，达到用户与资源解耦的目的。



本系统管理员是采用“分权管理”的方式，通过角色给管理员授权，每个角色对应不同权限，达到分权管理的目的。权限的控制除了在后台上做了限制，在前端页面上也做了控制，遵循“可见即可操作”的原则，管理员可操作他可见的页面或按钮，对没有操作权限的页面或按钮是隐藏的。

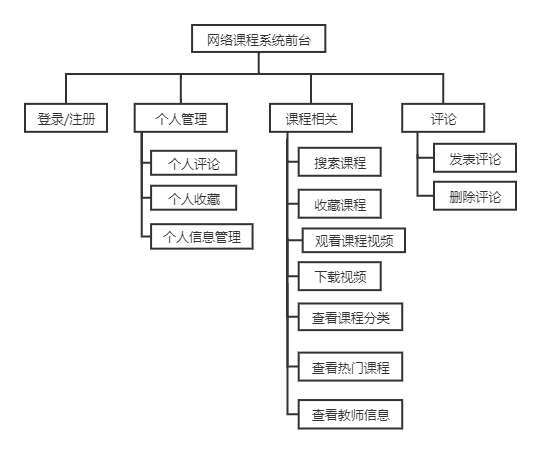
本系统用户分为四种角色，分别是普通用户、教师、普通管理员、超级管理员。其中超级管理员具备系统一切操作权限。普通管理员只具备所拥有角色对应的管理权限。教师拥有对应课程的管理权限。普通用户不具备后台管理权限，只具备慕课用户端页面的基本操作权限。

## 3.2 系统的功能需求

### 3.2.1系统功能说明

慕课系统分为用户端和管理端，用户端供普通用户自由免费学习视频课程，普通用户拥有登录注册、个人管理、搜索/收藏课程、在线学习/下载视频等基本功能。管理端有，本系统注重于后台数据的管理，用户端功能不会很多。

(1)用户端，为普通用户提供免费学习的视频课程，普通用户拥有登录注册、个人管理、搜索/收藏课程、在线学习/下载视频等基本功能。



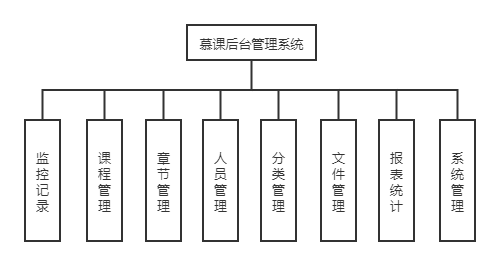
**图2 用户端功能结构**

（2）管理端

后台管理员具备监控记录、课程管理、人员管理、分类管理、文件管理、图表查看、系统管理等功能。分管和教师管理员只具备部分管理权限，对于分管不具备的管理权限在前端页面对应功能是隐藏的。

慕课后台管理，使得教师对视频得管理更便捷，还有管理员更方便对教师、普通用户进行管理，对教师的视频课程进行管理，可以删除违规用户、及课程，更改分类、系统的角色管理、在线人员管理、查看登录日志操作日志等等。具备报表权限的管理员还可以查看统计数据，统计数据以图表形式直观、简单地展示出来，让管理员可以知道系统的重要数据走势。

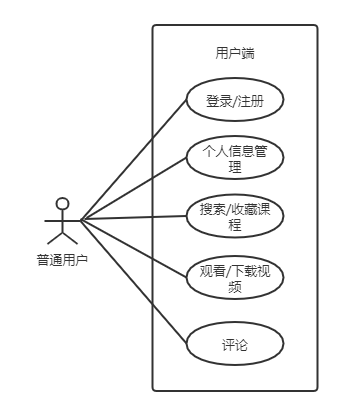
以下是后台管理系统的功能结构图。



**图2 管理端功能结构**

### 3.2.2系统用例图

本系统用户端的用例图如下：



管理端用例图

# 4系统概要设计

## 4.1系统功能结构图

系统的详细功能结构图如图1

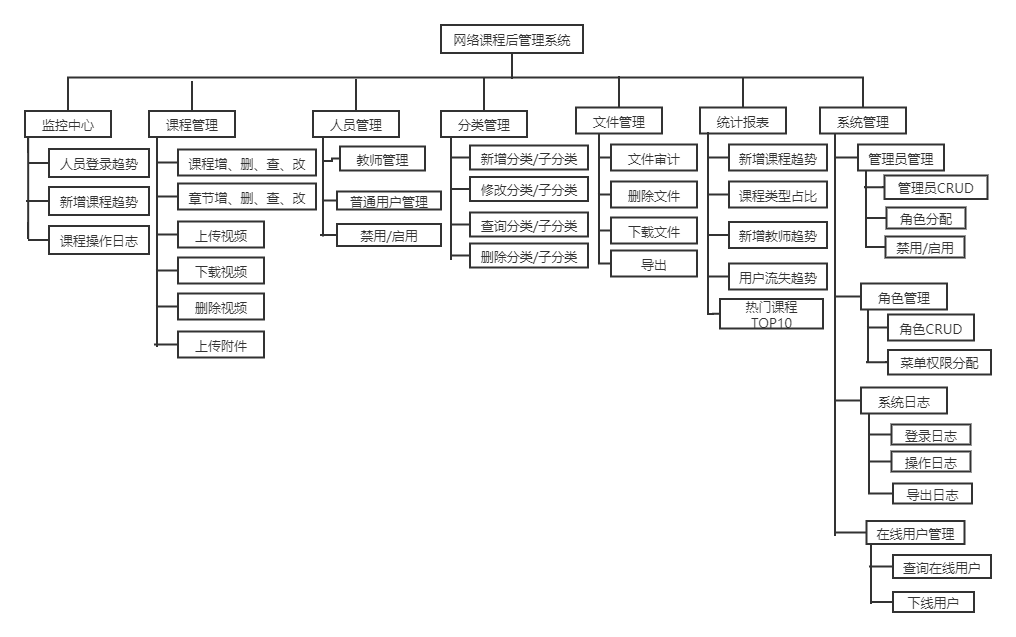


图1 系统详细功能结构

本后台系统的管理中心偏向于课程、人员、分类的管理，具备图表统计分析能力，如监控记录、统计报表等功能模块，都使得慕课的管理更直观。另外有附加功能，系统管理模块，包含管理员管理、权限管理、在线人员管理、系统日志等等。这些功能使得后台管理系统更灵活，更完整。

## 4.2系统的数据库设计

列举本系统重要模块/重要功能，一些具有一定代表性的表细述。

（1）用户表(mooc\_user)：用于记录本系统用户的id、姓名、头像、登录账号、登录密码等信息，本系统的普通用户和讲师均存这个表，由user\_type字段进行区分。具体信息如表1所示。

**表1 用户表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **允许空否** | **说明** |
| Id | bitint | 4 | 否 | 用户id，主键 |
| user\_image | varchar | 50 | 否 | 用户头像 |
| name | varchar | 40 | 否 | 用户昵称/姓名 |
| account | varchar | 20 | 否 | 登录账号 |
| password | varchar | 100 | 否 | 登录密码 |
| user\_type | varchar | 4 | 否 | 类型，教师/普通用户 |
| status | tinyint | 1 | 否 | 用户状态| 1正常，2禁用，3已删除 |
| create\_time | datetime | 1 | 否 | 创建时间 |
| update\_time | datetime | 1 | 否 | 更新时间 |

(2)系统管理员表(mooc\_manager)，用于本系统管理员的id、登录账号、登录密码等信息，如表2所示。

**表2 系统管理员表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **允许空否** | **说明** |
| id | bitint | 11 | 否 | 用户id，主键 |
| account | varchar | 20 | 否 | 登录账号 |
| password | varchar | 100 | 否 | 登录密码 |
| status | tinyint | 1 | 否 | 用户状态| 1正常，2禁用，3已删除 |
| create\_time | datetime |  | 否 | 创建时间 |
| update\_time | datetime |  | 否 | 更新时间 |

(3)课程表(mooc\_manager)，用于存储课程相关信息，如id、课程名、课程摘要简介、封面图片、学习人数等信息，如表3所示。

**表3 课程表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **允许空否** | **说明** |
| id | bitint | 4 | 否 | 课程id，主键 |
| name | varchar | 40 | 否 | 课程名 |
| teacher\_id | bigint | 11 | 否 | 教师id |
| summary | varchar | 2000 | 否 | 课程概述 |
| duration | int | 4 | 否 | 时长|单位秒 |
| image | varchar | 100 | 否 | 封面图片 |
| learning\_num | bigint | 11 | 否 | 学习人数 |
| comment\_num | bigint | 11 | 否 | 评论数 |
| status | tinyint | 1 | 否 | 状态|0草稿/1发布/2禁用/3已删除 |
| create\_time | datetime |  | 否 | 创建时间 |
| update\_time | datetime |  | 否 | 更新时间 |

(4)小章节表(chapter\_section)，用于记录课程对应的章节信息，如表4所示

**表4 小节表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **允许空否** | **说明** |
| id | bitint | 4 | 否 | 小节id，主键 |
| title | varchar | 50 | 否 | 标题 |
| course\_id | bigint | 11 | 否 | 所属课程id |
| chapter\_id | bigint | 11 | 否 | 所属章节id |
| video | varchar | 200 | 否 | 视频URI |
| file\_id | bigint | 11 | 否 | 文件表Id |
| duration | int | 4 | 否 | 时长|单位秒 |
| sort | bigint | 4 | 否 | 顺序 |
| create\_time | datetime |  | 否 | 创建时间 |
| update\_time | datetime |  | 否 | 更新时间 |

(5)文件表(mooc\_file)，用于记录本系统的所有类型的文件信息，包含视频、图片、ppt、pdf等文件。并且为了支持文件的分片上传、断点续传记录大文件的分片信息。如表5所示。

**表5 文件表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **允许空否** | **说明** |
| id | bitint | 4 | 否 | 文件id，主键 |
| name | varchar | 200 | 否 | 文件名 |
| file\_path | varchar | 400 | 否 | 文件相对路径 |
| file\_size | int | 11 | 否 | 大小|字节Byte |
| file\_suffix | varchar | 10 | 否 | 文件后缀 |
| file\_key | varchar | 100 | 否 | 文件唯一标识 |
| file\_type | int | 4 | 否 | 文件类型|1视频、2图片、3未知类型、4txt、5markdowm 5、ppt 6 word 7、excel 8、pdf |
| shard\_index | int | 4 | 否 | 文件当前分片下标 |
| shard\_count | int | 4 | 否 | 总共分片数 |
| shard\_size | int | 4 | 否 | 分片大小 |
| user\_id | bigint | 11 | 否 | 所属用户ID |
| course\_id | bigint | 11 | 否 | 所属课程ID |
| status | tinyint | 1 | 否 | 文件状态| 1正常，2已删除 |
| create\_time | datetime |  | 否 | 创建时间 |
| update\_time | datetime |  | 否 | 更新时间 |

(6)菜单权限树表(menu\_tree)，用于记录本系统的菜单及系统接口的访问权限。本系统对所有的访问都进行权限控制，菜单均由后台存储，前端根据后台返回的菜单信息展示对应的页面。如表6所示。

**表6 菜单权限树 表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **长度** | **允许空否** | **说明** |
| id | bitint | 4 | 否 | id，主键 |
| label | varchar | 20 | 否 | 菜单标签|名字 |
| menu\_key | varchar | 48 | 否 | 菜单唯一标识 |
| icon | varchar | 48 | 否 | 菜单图标 |
| parent\_id | varchar | 48 | 否 | 菜单父ID|0表示根节点 |
| permission | varchar | 255 | 否 | 权限 |
| router | varchar | 255 | 否 | 页面路由 |
| leaf | tinyint | 1 | 否 | 是否叶子节点|0不是，1是 |
| create\_time | datetime |  | 否 | 创建时间 |
| update\_time | datetime |  | 否 | 更新时间 |

## 4.3系统流程图

# 5 系统详细设计与实现

## 5.1系统组成

本系统由后端7个微服务加前端工程组成，后端七个微服务分别是 核心服务、权限服务、统计报表服务、文件服务、网关服务、注册中心服务。每个微服务具备不同职责，提供对应的功能实现。

**核心服务：**系统的主体服务，提供大部分业务的视图展示的接口，主要负责和前端页面对接，主要功能有课程管理、章节管理、分类管理、前台页面等。

**文件服务：**提供文件的上传和下载、视频的播放、文件预览等服务。

**权限服务：**负责系统的登录授权和验证，其功能有登录、注册、管理员管理、角色管理、权限管理、在线人员管理等。

**统计报表服务：**负责统计系统中的数据，主要功能体现在监控中心的趋势图，以及报表管理统计的图表。

**网关服务：**网关服务，统一前端请求的入口，负责路由转发不同请求到对应的服务。

**注册中心服务：**注册中心，每个微服务都需要向注册中心注册，方便统一管理和监控服务状况。

**前端工程：**

前后分离项目中，前端是单独一个工程，主要负责对接后端接口，展示对应视图和数据。

## 5.1系统配置

### 5.1.1后端框架搭建

后端七个微服务都是一个个单独的SpringBoot工程，每个微服务具备自己的独立的配置文件。为了方便开发、部署、测试，区分环境运行，配置文件有选择加载环境的配置文件bootstrap.yml、开发环境配置文件application-dev.yml、测试环境配置文件application-qa.yml、本地运行的配置application-local.yml。本地、开发、测试三个环境基本相同，只是配置的组件机器ip不一样，下面的配置都以local配置为例。

代码1为bootstrap.yml的配置，bootstrap.yml文件会在SpringBoot启动前会比application.yml先加载，常用于动态选择加载哪个环境的配置。

代码1 bootstrap.yml配置

|  |
| --- |
| #选择加载开发环境dev、测试环境qa、本地环境local  spring:  profiles:  active: local |

代码2为各个服务共同的配置，除了配置端口和服务名，还要配置注册中心。不同服务的端口和服务名称不一样，这里以核心服务为例子。

代码2 bootstrap.yml重要配置代码

|  |
| --- |
| #配置服务端口  server:  port: 9001  # servlet:  # context-path: /core  #向eureka注册中心注册  eureka:  client:  service-url:  defaultZone: http://localhost:8888/eureka/  #配置服务名  spring:  application:  name: mooc-core |

### 5.1.2数据库的配置

本系统使用到三种数据库，分别是关系型数据库MySQL、基于内存的非关系型数据库Redis、还有优秀的检索和分析引擎数据库Elasticsearch。

数据库在SpringBoot使用这三种数据库都非常简单，只需要引入对应的pom依赖，再通过简单的配置就可以实现。本系统使用mysql使用的是jpa,使用SpringBoot默认的连接池HikariCP来管理数据库连接。

代码2为各个数据库的配置

**代码3 数据库配置**

|  |
| --- |
| spring:  #es连接配置  data:  elasticsearch:  cluster-nodes: 127.0.0.1:9300  cluster-name: my-application  # mysql数据库连接配置  datasource:  driver-class-name: com.mysql.cj.jdbc.Driver  url: jdbc:mysql://localhost:3306/mooc?characterEncoding=UTF-8&serverTimezone=GMT%2B8  username: root  password: root  #redis数据库配置  redis:  host: localhost  port: 6379  database: 0  password: root  port: 6379 |

### 5.1.3 微服务session共享的实现

不同微服务之间无法通过session共享登录信息，通过系统颁发token（令牌），利用redis第三方缓存数据库redis来共享登录信息，实现单点登录的功能。

### 5.1.4手机短信通知

### 5.1.5导出Excel文件

## 5.3系统安全

常见系统攻击手段有DDoS攻击、通过抓包工具抓取登录请求获取账号和密码、SQL注入、XSS等等。为了系统安全，本系统针对上述攻击手段做必要的防护措施，使得系统的安全性大大提升。

**DDoS攻击攻击防护**

DDoS攻击又叫分布式拒绝服务攻击，是一种非常恶意并且粗暴的攻击手段，简单来说就是攻击者通过利用网络协议漏洞多台机器同时向系统某些接口发送大量请求，不断消耗网络带宽和系统资源最终导致系统瘫痪不可用。

DDoS攻击是很难完全防护，针对DDoS攻击，本系统采用的防护策略是使用Nginx限制每个IP分钟内请求数加上IP黑名单的方式，可以一定程度上抵御DDoS攻击。

**SQL注入防护**

尽量使用Mybatis、Jpa等ORM层框架，能大大降低sql注入的风险，使用Mybatis时参数占位符‘#’ 符号，而不要是用$。禁止SQL拼接方式的代码，若使用拼接一定要使用占位符号‘?’作为参数占位符。所有输入参数必须经过后台校验，过滤非法字符。这样几乎可以杜绝sql注入的风险。

**设置安全的Http响应请求头**

在Nginx统一设置响应头的X-XSS-Protection、X-Content-Type-Options、Content-Security-Policy等，能有效防御XSS(跨站脚本攻击)，提高系统的安全性。

**敏感信息加密传输**

在登录、注册、或者修改密码等功能中，可能涉及到密码、手机号等敏感信息的传输，为了保证这些敏感信息在网络传输过程中不被获取非法分子截取，敏感信息统一采用RSA算法加密。前端保存公钥，使用公钥加密敏感信息，后台接收到加密内容通过私钥解密。传输过程中即使网络传输报文被不法分子截取了，因为没有秘钥解密，也无法获取里面的内容，以此来保证了用户信息的安全。

**口令加密存储**

系统的密码一般是采取加密方式去存储的，目的是为了防止由于sql注入获取其他攻击导致系统数据库信息泄露。若不采取加密方式存储，在数据库被泄露情况下，不法分子很容易通过密码登录系统，进行非法行为操作。

为了确保用户密码的安全，本系统使用的是Bcrypt工具进行密码加密。Bcrypt是非对称加密的算法，与MD5比较其安全保障在于其加密和解密速度缓慢，大约是MD5加密算法的速度相差上百万倍，解密速度慢使得非法人员很难利用计算机的计算速度进行暴力破解，简单来说破解密码的代价很高。而且Bcrypt对于每次加密都是使用随机盐的方式，即使某个密码被破解了，其他密码仍然安全。

**账号管理**

超级管理员默认不允许修改，由后台配置。普通管理员账号由超级管理员分配。合作伙伴账号由普通管理员在定义合作伙伴时分配。账号在整个系统里面是唯一的，长度为3到16位字母或数字的组合。系统支持针对每一个系统的用户都分配一个账号，杜绝共享账号。

**日志记录**

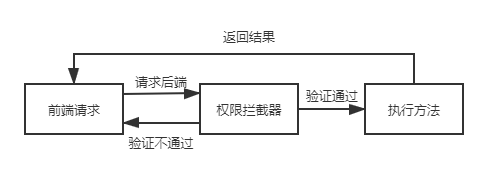
为了保证后台管理系统的操作安全性，管理员的登录信息和操作信息都会被记录到系统管理模块的登录日志和操作日志中，包括操作人、操作实际、操作IP, 保证所有的修改记录都有迹可寻。超级管理员或有日志权限的管理员可以阅览。留痕记录的目的也是为了监控管理员的操作或者误操作，防止无意义的纠纷。其实现的方式是使用Spring AOP，使用切面的的技术对系统日志进行记录。AOP面向切面编程是一种代码入侵性低的增强编程思想，可用用到日志记录功能里。

## 5.3系统目录结构

## 5.4系统功能实现

### 5.3.1监控记录功能模块

授权验证流程



### 5.3.2课程管理功能模块

### 5.3.3章节管理功能模块

### 5.3.4用户管理功能模块

### 5.3.5报表统计功能模块

### 5.3.6文件管理功能模块

### 5.3.7系统管理模块

### 5.3.7管理员管理功能模块

### 5.3.8角色管理功能模块

### 5.3.9在线人员管理功能

### 5.3.10个性化设置功能

# 6 系统的测试与分析

单元测试和集成测试，采用的是TDD(测试驱动开发)的方式，使单元测试的覆盖面更广。接口测试采用的使自动化测试的方式，减少频繁修改频繁回测的问题，大大减少人力成本。黑盒测试主要应用等价类划分、边界值分析等方法，提前编写系统的测试用例，全面覆盖系统的功能，保证在需求场景下系统质量。另外性能压测是对本系统调用数比较大的接口进行压测，要求100线程tps不低于100。最后的渗透测试是为了发现系统隐含的安全漏洞，这些安全漏洞都可能被不法分子恶意攻击。

## 6.1系统测试的目的和原则

### 6.1.1系统测试的目的

软件测试的目的一般为了发现程序中隐含的错误，这些错误可能是程序开发人员在变成过程中考虑不周全导致。软件测试就是为了尽可能发现这些缺陷，避免造成更大的损失。软件测试是软件开发过程中不能缺少的一个环节，也是保证质量的重要检验。

### 6.1.2系统测试的原则

软件测试有以下几个原则：

1. 一切测试用例都应该符合用户需求，因为软件开发就是根据需求而进行编码的，如果不符合需求那么就是属于缺陷。
2. 软件测试计划应该在软件开发的初始阶段就着手制定，提前编排好测试的阶段和时间，提前针对需求功能编写测试用例。
3. 一般先从小规模测试开始，后面逐渐转向大规模的测试，从单元测试逐渐转集成测试。
4. 软件测试必须要有专门的第三方专业的测试人员来做，不能只依赖于开发人员的单元测试和自测。

## 6.2系统测试阶段

为了保证系统的质量，本系统进行五个阶段的测试：

第一阶段是开发阶段的单元测试和集成测试，这一部分是由开发人员进行的，保证单元、模块程序正确性。

第二阶段是在系统开发完成后进行的功能测试，由测试人员进场根据用例进行测试，一般是黑盒测试。

第三阶段是性能压测阶段，使用Jmeter工具对系统一些“高频”接口进行压力测试，模拟多个用户并发使用的场景，根据结果来评判系统的性能表现。

第四轮是保证系统安全的渗透测试，由专门的渗透测试人员进行，目的是寻找系统可能存在的安全漏洞。

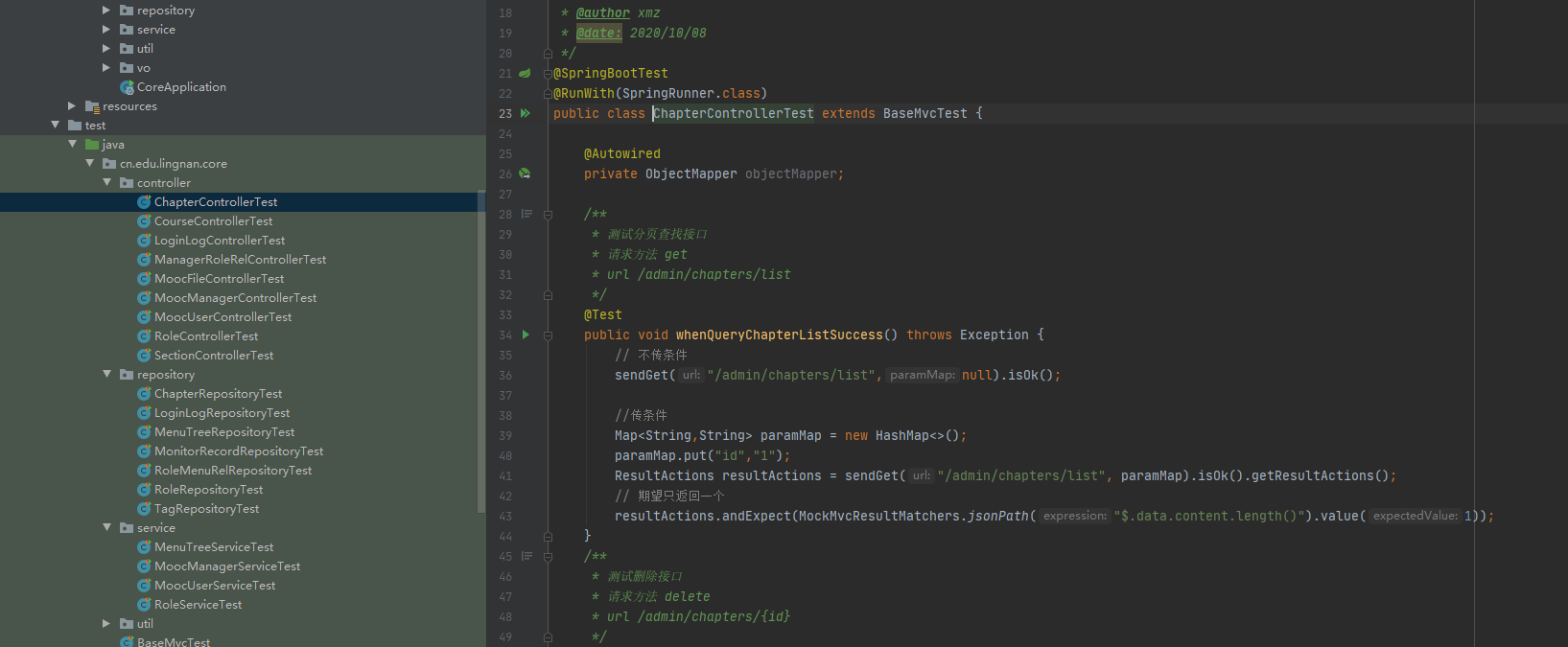
最后是验收阶段，一般由产品和测试人员一起进行系统功能的体验，确保系统流畅页面功能完整，符合需求和预期。这时整个软件开发、测试流程已经经接近尾声。

经过这五轮的测试来保证系统功能的完整性和安全性、健壮性。

## 6.3对本系统的测试

对本系统的测试也是严格按照以上测试阶段来做的。

（1）进行单元测试及集成测试，由开发人员开发阶段，对每一方法、模块进行测试，使用的是junit测试框架。



（2）功能测试，由测试人员进行，在开发完成后按照提前写好的测试用例，在页面上进行测试。这个阶段使用的是黑盒测试，测试人员在不知道功能内部具体逻辑，按照需求文档，功能输入输出对系统功能进行全面的测试。

（3）压力测试，使用压测工具Jmeter模拟大量用户访问，观察数据的正确性和系统的稳定性，压测是对系统性能的重大考验，压力测试有两个重要指标TPS(Transactions Per Second)和QPS(Queries Per Second)，分别指每秒可以处理事务数目和查询数目。一般来说，TPS和QPS代表系统应对并发能力越好，可以作为确保系统交付质量的保证。

（4）渗透测试，通过各种工具，模拟黑客可能使用的攻击手段，抓住技术漏洞，对系统进行全面的检查，尽可能发现存在的安全漏洞，防范于未然，以免系统上线后发生严重损失。

最后小结，本系统经过以上几轮测试，发现了很多软件缺陷，但是后面这些都已经修复。缺陷修复也是软件开发的一个流程，往往经过测试才能及时被发现，所已软件测试是一个很重要、不可或缺的一个环节。

# 7 总结与展望

初衷

本系统是基于微服务Spring Cloud框架的后台管理系统，做微服务的初衷也是为了可以学习到更多技术组合。在这个过程中除了学习到后台的微服务架构和一系列后端技术栈，还学习到前端的Vue.js、Element-UI等框架的使用，并且部署运维能力也有所提升.这过程中踩了很多坑，同时也学习到了很多知识点。

总结一下，我选择微服务架构开发的原因主要有以下三点:

1、在工作中使用到，希望毕设可以加强自己的职业能力。

2、可以灵活选择不同技术组合，比如ORM层框架，在有些服务使用的是Jpa，在权限服务使用的是Spring JDBC，部分服务使用国内较欢迎的Mybatis。不同服务是属于不同工程，他们代码是完全隔离的，不要求统一技术栈。

3、方便以后后续扩展和学习微服务部署相关知识。 前面也说过微服务把一个工程拆开成多个工程，那部署、管理、监控的难度也会提升。后续可以沿着这条线学习jenkins自动化部署工具，还有docker、[Kubernetes](https://www.kubernetes.org.cn/" \t "_blank)等容器化部署技术。

**不足和改进：**

本系统虽然基本功能已经实现，但不足之处比较多，设想与现实不一样，比如评论回复模块、消息通知模块本来在设想中，后面没有时间去弄。计划的技术栈里的RabbitMQ消息队列也没用上。软件开发本身就是个复杂的过程，风险难以把控，本系统是从去年8月份就开始，至今跨度6个月，虽然横跨的时间很长，但是实际开发时间其实并不多。都是利用下班零星的时间拼凑，开发效率不高。

后续的优化，首先是缺失功能的填补，对于评论模块、消息通知模块后续会通过新增一个微服务的方式，无入侵地、不影响其他模块地新增。二是需要美化页面，特别是涉及到图表那一块，现在还不是很好看需要继续调整。三是架构的调整，我对这个初版的微服务架构还不是很满意，职责划分不够明确，工程代码还没有完全物理分离这些还需要调整。最后一点是需要对整个系统进行性能优化和代码重构，目前只针对功能的实现，还没很好地处理高并发、大数据量的情况。高可读可读性、高质量的代码也不是一次就能写出来的，往往需要不断重构，不断优化。

**最后总结：**

在这过程中印象最深的是权限服务的代码编写，写了三种不同的实现方式，使用到的权限技术框架不一样。一个是Spring Security, 一个是使用Apache Shrio,和不使用任何权限框架自定义拦截器+自定义注解实现对方法级别的权限控制。不同框架可能语法不太一样，但是其实实现的都是做同一件事，解决同样的问题。在对比和学习中可以学到很多知识。

做毕设过程中，也学习到了Elasticsearch这个优秀的搜索引擎，更加灵活使用Redis、Mysql。学习到基于RBAC的权限模型，熟悉了Spring Security、Shrio等权限框架。在这过程中不是一帆风顺，遇到过许多技术性难点，但是通过搜查资料、谦虚向他人请教等途径，这些技术问题都被一一解决。

做好一件事是比较艰难的，软件开发也不例外，开发这个慕课后台管理系统一个人几乎顶了一个团队。从需求调研、功能规划、架构设计、技术选择、数据库设计、开发编码、测试、文档编写、运维部署，这些工作都得一个人独立完全了。工作量巨大，自己充当了项目经理、业务经理、架构师、数据库工程师、后端开发、前端开发、测试人员、运维人员。总的来说，除了体验软件开发的艰辛过程，还对各个IT岗位的工作职责内容都有了更深的理解，对自己的能力有了很大的提升。最后对于作为开发人员的我来说，还掌握了发现问题、思考问题和解决问题的能力，我相信，这些在我以后的职业生涯都是宝贵的经验。

**参考文献**

[1]关德君,林丽.互联网时代——慕课与传统教育的比较研究[J].湖北开放职业学院学报,2021,34(04):28-29.

[1]吴大非.MOOC管理平台的设计与开发[J].电脑知识与技术,2018,14(27):47-49+52.

[1]赵夫群,王静,宋晓俊.中小学MOOC动态学习资源网站的开发[J].科技风,2019(08):31.

[1]Galikyan Irena and Admiraal Wilfried and Kester Liesbeth. MOOC discussion forums: The interplay of the cognitive and the social[J]. Computers & Education, 2021, 165(prepublish) : 104133-.

[1]喻梅,王建荣,于健,徐天一,赵满坤,高洁,刘志强.基于MOOC的线上线下混合教学实践[J].计算机教育,2021(03):1-5.

[1] 马克思．关于《工资、价格和利润》的报告札记[M]//马克思，恩格斯．马克思恩格斯全集：第44卷．北京：人民出版社，1982：505．

[2] 钟文发.非线性规划在可燃毒物配置中的应用[C]∥赵玮．运筹学的理论与应用：中国运筹学会第五届大会论文集.西安:西安电子科技大学出版社，1996：468-471.

[3] WEINSTEIN L,SWERTZ M N．Pathogenic properties of invading microorganism [M]//SODEMAN W A, Jr.，SODEMAN W A．Pathologic physiology: mechanisms of disease．Philadelphia: Saunders, 1974:745-772.

[4] 中国地质学会.地质论评[J]. 1936,1(1)-.北京：地质出版社，1936-.

[5] 中国图书馆学会.图书馆学通讯[J]. 1957(l)-1990(4).北京：北京图书馆，1957-1990.

[6] American Association for the Advancement of Science．Science [J].1883,1(1)—. Washington, D.C.：American Association for the Advancement of Science, 1883-.

[7] 李晓东，张庆红，叶瑾琳.气候学研究的若干理论问题[J].北京大学学报：自然科学版，1999，35(1)：101-106.

[8] 刘武，郑良，姜础.元谋古猿牙齿测量数据的统计分析及其在分类研究上的意义[J].科学通报，1999，44(23)：2481-2488.

[9] 莫少强.数字式中文全文文献格式的设计与研究[J/OL]．情报学报，1999，18(4):1-6[2001-07-12]．http://eriodical.anfangdata.com.cn/periodical/gb xb/qbxb99/qbxb9904/990407. htm.

[10] 姜锡洲．一种温热外敷药制备方案:中国，88105607.3[P]．1989-07-26.

[11] 西安电子科技大学.光折变自适应光外差探测方法:中国，01128777.2[P/OL]. 2002-03-06[2002-05-28]. http://211. 152. 9.47/sipoasp/zljs/hyjs-yx-new. asp? recid=01128777. 2&Ieixin=0.

**参考文献详细说明**

参考文献置于正文的末尾；参考文献序号用[1]、[2] ……表示，且序号左顶格，每一条参考文献著录均以“.”结束。依据**GB/T 7714—2005（文后参考文献著录规则2005年版）**，参考文献可区分为直接参考文献（即引文注释）和间接参考文献。当既有引文注释,又有间接参考文献时,直接参考文献（即引文注释）在前,间接参考文献在后。

参考文献的著录格式应符合中华人民共和国国家标准GB/T 7714—2005,即文后参考文献著录规则2005年版。在著录过程中正确使用文献类型标志代码和电子文献载体标志代码，详见表3和表4。

**表4 参考文献类型和标志代码**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **参考文**  **献类型** | 普通  图书 | 会议  录 | 汇  编 | 报纸 | 期  刊 | 学位  论文 | 报  告 | 标  准 | 专  利 | 数据  库 | 计算机  程 序 | 电子公告 |
| **标志代码** | M | C | G | N | J | D | R | S | P | DB | CP | EB |

**表5电子文献载体类型和标志代码**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **电子文献载体类型** | 联机网上数据库 | 磁带数据库 | 光盘图书 | 磁盘软件 | 网上期刊 | 网上电子公告 |
| **标 志 代 码** | DB/OL | DB/MT | M/CD | CP/DK | J/OL | EB/OL |

常用参考文献类型的著录格式如下：

**1. 参考文献是专著的著录格式**

主要责任者．题名：其他题名信息[文献类型标志]．其他责任者．版本项．出版地：出版者，出版年：引用日期．获取和访问路径．

示例：

引用日期

出版者

出版年

出版地

主要责任　　　　者

文献类型标志

题名

[1] 赵耀东．新时代的工程师[M/OL]．台北：天下文化出版社，1998[1998-09-26]． http：//www. Ie.nthu.edu.tw/info/ie.newie.htm(Big5).

获取和访问路径

**２.参考文献是专著中的析出文献[[1]](#footnote-1)的著录格式**

析出文献主要责任者．析出文献题名[文献类型标志]．析出文献其他责任者∥专著主要责任者．专著题名：其他题名信息．版本项．出版地：出版者，出版年：析出文献的页码．获取和访问路径．

示例:

[1] 马克思．关于《工资、价格和利润》的报告札记[M]//马克思，恩格斯．马克思恩格斯全集：第44卷．北京：人民出版社，1982：505．

[2] 钟文发.非线性规划在可燃毒物配置中的应用[C]∥赵玮．运筹学的理论与应用：中国运筹学会第五届大会论文集.西安:西安电子科技大学出版社，1996：468-471.

[3] WEINSTEIN L,SWERTZ M N．Pathogenic properties of invading microorganism [M]//SODEMAN W A, Jr.，SODEMAN W A．Pathologic physiology: mechanisms of disease．Philadelphia: Saunders, 1974:745-772.

**３.参考文献是连续出版物的著录格式**

主要责任者题名：其他题名信息[文献类型标志]．年，卷（期）-年，卷（期）出版地：出版者，出版年．获取和访问路径．

示例：

[1] 中国地质学会.地质论评[J]. 1936,1(1)-.北京：地质出版社，1936-.

[2] 中国图书馆学会.图书馆学通讯[J]. 1957(l)-1990(4).北京：北京图书馆，1957-1990.

[3] American Association for the Advancement of Science．Science [J].1883,1(1)—. Washington, D.C.：American Association for the Advancement of Science, 1883-.

**4．参考文献是连续出版物中的析出文献的著录格式**

析出文献主要责任者．析出文献题名[文献类型标志]．连续出版物题名：其他题名信息，年，卷(期)：页码[引用日期]．获取和访问路径．

示例：

[1] 李晓东，张庆红，叶瑾琳.气候学研究的若干理论问题[J].北京大学学报：自然科学版，1999，35(1)：101-106.

[2] 刘武，郑良，姜础.元谋古猿牙齿测量数据的统计分析及其在分类研究上的意义[J].科学通报，1999，44(23)：2481-2488.

[3] 莫少强.数字式中文全文文献格式的设计与研究[J/OL]．情报学报，1999，18(4):1-6[2001-07-12]．http://eriodical.anfangdata.com.cn/periodical/gb xb/qbxb99/qbxb9904/990407. htm.

**5．参考文献是专利文献的著录格式**

专利申请者或所有者．专利题名：专利国别，专利号[文献类型标志]．公告日期或公开日期[引用日期]．获取和访问路径．

示例：

[1] 姜锡洲．一种温热外敷药制备方案:中国，88105607.3[P]．1989-07-26.

[2] 西安电子科技大学.光折变自适应光外差探测方法:中国，01128777.2[P/OL]. 2002-03-06[2002-05-28]. http://211. 152. 9.47/sipoasp/zljs/hyjs-yx-new. asp? recid=01128777. 2&Ieixin=0.

**６．参考文献是电子文献[[2]](#footnote-2)的著录格式**

主要责任者．题名：其他题名信息[文献类型标志/文献载体标志]．出版地：出版者，出版年（更新或修改日期）[引用日期]．获取和访问路径．

示例：

[1] PACS-L：the public-access computer systems forum[EB/OL]．Houston，Tex：University of Houston Libraries，1989[1995-05-17]．http://info.lib. uh. edu/pacsl. Html．

[2] Online Computer Library Center, Inc. History of OCLC[EB/OL].[2000-01-08]. http: //www. oclc. org/about/history/default. html．

**附 录**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××。

**致 谢**

××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××。

1. 从整本文献中析出的具有独立篇名的文献,如从期刊中析出的一篇论文。 [↑](#footnote-ref-1)
2. 电子文献是以数字方式将图、文、声、像等信息存储在磁、光、电介质上，通过计算机、网络或相关设备使用的记录有知识内容或艺术内容的文献信息资源，包括电子书刊、数据库、电子公告等。 [↑](#footnote-ref-2)