

**本科毕业论文（设计）**

[基于B/S模式的精品课程网站设计与实现](javascript:showViewModel(1,'1328494488397815809'))

|  |  |
| --- | --- |
| **学院** | **计算机工程学院** |
| **专业** | **软件工程** |
| **年级** | **2017级** |
| **学号** | **192017145370** |
| **姓名** | **胡号南** |
| **指导师** | **孔令信** |

**2020 年 2 月**

**基于B/S模式的精品课程网站设计与实现**

**摘要：** 在21世纪的今天，随着社会的不断发展与进步，计算机应用技术得到了快速的普及，互联网早已渗透于各个行业之间，深刻地影响着大众的日常生活，为我们的学习和工作带来了极大的便利。在这一时代背景之下，教育行业也走向了数字化的变革，数字化的教育平台终也迎来了全新的发展机遇。如实现了对教育的数字化改革，由原线下教育转变成线上教育，教学将不再受到时间与空间的影响，对于普通大众来说将成为一种更加平等的社会资源。

本论文主要是对基于B/S模式的精品课程网站系统进行了介绍，包括研究现状，涉及的开发背景，对系统的设计目标进行了阐述，并涵盖系统的需求分析，以及整个系统的设计方案，对本系统的设计以及实现都有比较细致的论述，最后包括了对该系统性能以及使用的一些具体测试。

本系统包括了前端与后端两部分，前端以Vue.js框架与ElementUI组件库对前端页面进行设计与开发，后端以Java为语言核心，使用SpringBoot框架并于其它主流技术栈相结合对后端逻辑进行设计与开发，持久层采用了MySql数据库作为数据支撑。本系统实现了数字化的网络课程平台，设计了一个高校网络课程管理系统，其主要使用者分为教师、学生、管理员三个角色。主要功能包括课程录播、课程直播、课程作业分发与提交并自动计算成绩、课程成绩统计与排行等主要功能，实现了对教育资源的高效管理。

**关键词：**线上教育；B/S模式；精品课程网站；网络课程平台

**Design and Implementation of**

**Excellent Course Website Based on B/S Mode**

**Abstract：**In today’s 21st century, with the continuous development and progress of society, computer application technology has been rapidly popularized, and the Internet has penetrated into various industries, profoundly affecting people’s daily life, and bringing great benefits to our study and work. convenient. In the context of this era, the education industry has also undergone digital transformation, and the digital education platform has finally ushered in new development opportunities. If the digital education reform is realized and the original offline education is transformed into online education, teaching will no longer be affected by time. And space, and will become a more equal social resource for the public.

This article mainly introduces the excellent course website system based on the B/S model, including the research status, the development background involved, expounds the design goals of the system, and covers the system requirements analysis and the design of the entire system, the system design and Implementations are discussed in more detail, regarding the final characteristics of system performance and the use of some specific tests.

The front-end uses Vue.js framework and ElementUI component library to design and develop front-end pages; the back-end uses Java as the language core, and SpringBoot framework and other mainstream technology stacks to design and develop back-end logic. The system implements a digital network course platform and designs a university network course management system. Its main users are divided into three roles: teacher, student, and administrator. The main functions include: course records, course live broadcast, course distribution and submission, automatic score calculation, course score statistics and ranking, etc., which realize the efficient management of educational resources.

**Key words:** Online education; B/S mode; [High-quality](javascript:;) [curriculum](javascript:;) [website](javascript:;); Online course platform

# 一、概述

## （一）选题背景及意义

近年来，学校学生数量和教学资源逐渐增加，传统教学的方式已经不能满足现代化的需求，且教育部在《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》中强调精品课程建设是教学质量和教学方式改革的重要组成部分。线上教育的出现，成功解决了教育行业面临的几大痛点，如招生成本高、口碑传播效率低、学院流失量大等多个问题，这也是很多教育机构向线上发展的原因。为了更好的适应信息时代的高效性，一个利用计算机来实现学校精品课程网教学管理工作的系统将必然诞生。基于这一点，设计了一个精品课程网站，用来就学生精品课程网教学进行管理，以便在最短的时间内，高效准确的完成整个教学程。

基于对上述的认识，收集相关资料和数据，查阅有关文献及技术参数，对学校的实际需求等方面进行了调研，发现目前所采用的手工记录的方法进行管理存在以下几个方面的弊端:对象范围广，数据存储不易;填写的数据多，且大量数据重复;不易存档，成千上万的信息和堆积如山的单据对管理人员来说是一个负担，需要大量的文档资料。鉴于以上种种原因，学校教学的管理急需一种软件来代替手工操作来管理数据资料。

计算机技术在现代管理中的应用，使计算机成为领导者和管理人员应用现代技术的重要工具。因此，本研究尝试这一领域，建立一个基于B/S模式，并能实现操作简单，易管理、易维护，具有良好的共享性和交互性等特点的自助式精品课程网络平台，力求开发出适合我国教师使用的精品课程网络平台，促进这一领域中理论研究与技术实践的共鸣与发展。

随着高校信息系统的不断完善，学校的管理越来越依赖于信息化的管理。所以系统开发将围绕学校教学工作的实际情况，使之能迅速适应学校教师及学生的需要。在确切了解我国高校教师在精品课程建设方面的困难和需求的基础上，开发一套简单易用的精品课程网络平台，以使得教师能够在此平台上轻松的实现精品课程的建设、修改、更新，有利于充分调动教师建设精品课程的积极性。其实现的实际意义：减少教师的工作量；学生及教师可以随时浏览查看，且更加直观；信息化对课程的管理更加快捷。

## （二）国内外研究现状

随着互联网技术和多媒体技术的飞速发展，在线教育组件成为多媒体教学中的重要形式，它有着传统课堂教育不可比拟的优势，能把真实的课堂带到用户面前。在线课程以其多样性、直观的特性，吸引了越来越多的人的注意，成为开放课程运动中一股不可忽视的力量。我国的网络教育始于1994年国家实施的教育科研网示范工程。网络教育在我国已经有了一定的发展规模，虽然发展趋势缓慢，但任保持上升势态。目前国内存在众多在线教育平台，其涵盖的专业丰富，师资力量强大，课程种类繁多，但课程开放程度不足，专业性不强，各个专业与高校开放的课程资源数量参差不齐。放眼国外，全球已有800多所大学通过互联网提供网上学位课程，如哈佛大学、麻省理工学院、剑桥大学都允许国外学生进行网上注册、异地学习学位课程。国外以自主学习、点对点与广播式教学为根基思想，但在在线答疑功能方面不足、市场利益驱动下使得课程费用高昂等方面体现了不足之处。总而言之，在线教育的普及性远未达到人们对它的需求，需要通过更加正规化，标准化，法制化，以优质的服务来提高教育的质量。

## （三）本课题主要工作

一开始，本文就对系统内谈到的基本知识，从整体上进行了大致的描述，以下内容将在此基础之上进行系统的分析。为了能够使本系统较好、较为完善的设计出来，就需要先进行分析调查。基于之前相关的基础，在功能上对系统进行了概括描述，然后通过详细的分析进一步进行系统的设计，其次探讨系统的具体功能需求，研究其可行性，梳理系统设计，以及设计之后的成功展示，和相关测试数据的展示，最后是对项目进行完善并发布。

# 二、系统可行性分析

系统的可行性分析的目的就是用最小的代价在尽可能短的时间内确定问题是否能够解决，是为了确保系统能够具备开发和运行的必备条件比不缺少的重要环节。精品课程网系统的主要目标是为了共享教育资源，在确定了这一目标之后，我们可以从技术、操作、经济以及法律，这四大方面进行详细的阐述。

## （一）技术可行性

精品课程网系统是基于前后端分离开发模式，前端使用Vue.js框架以及ElementUI组件库相结合开发，后端采用三层架构的思想且以Java编程语言为基础的SpringBoot框架与Shiro安全框架、ElasticSearch搜索引擎、RabbitMQ消息中间件、MyBatis-Plus持久层框架、JWT授权框架、Druid连接池、Swagger文档生成测试集成工具、以及部分阿里云API相结合进行开发，持久层使用MySql数据库以及Redis缓存数据库，项目最终以Nginx反向代理服务器与Docker容器引擎相结合进行部署。

## （二）操作可行性

操作可行性分析主要是指用户使用的舒适性，因此界面设计时需要考虑使用者的操作习惯。精品课程网使用B/S架构，通过浏览器来实现对系统的交互，在使用便捷方面能达到简易的效果。系统的页面采用Vue.js框架完成，其反馈速度快、后端使用Nginx服务器，能够使得数据能够在第一时间内发送到客户端，可以提供给用户流畅的使用体验。在系统的UI设计方面，页面的交互设计的极其简单，可以通过少量的操作达到极高的效率，页面的样式设计以现代化、年轻化、简洁化的基底呈现给用户。在智能设备普及的今天，用户使用本系统的成本是极其低廉的，由此可见本系统的操作是完全可行的。

## （三）经济可行性

精品课程网系统是基于B/S模式，不依赖第三方商店平台，且采用的所有研发技术都是以开源免费的性质提供，几乎不计算成本。其所要求的硬件和软件环境，在市场上都很容易购买，程序开发主要是在系统的开发和维护。所以程序在开发人力、财力上要求不高，且系统的开发周期短、设计简单、维护便捷。相对于传统的教育模式而言，其所需要的人力资源、社会资源、自然资源都远远低廉。如果采用线上教育的模式，所需要开销的成本都可以得到很好的控制。由此可见，此系统在经济方面有着较高的可行性。

## （四）法律可行性

法律可行性主要考虑体统是否存在任何隐私侵犯、妨碍和责任问题。精品课程网是由本人所设计完成，在技术选择方面都使用了开源项目，并遵循其开源的协议。在网站内容方面，所有网站内容都出自于作者本人，或出自于互联网上已得到授权使用的资源，并完全遵守中国法律。所以这个系统的开发、设计以及内容上，并不会存在侵权等问题，在法律上是完全具有可行性的。

# 三、系统分析

## （一）需求分析

需求分析是软件计划阶段的重要活动，也是软件生命周期的重要环节。需求分析讨论的是系统在功能上需要实现什么类似的功能性需求，也涵盖软件的一些非功能性需求，从而使得软件后续的开发达到既定的目标。

### 1**．**功能需求分析

通过前期的了解，基于B/S模式的精品课程网站（以下简称精品课程网）主要使用的人群包括但不限于；在校学生、高校教师、及其它有学习需求的群体。将使用群体抽象化可分为两大类；拥有教育资源的教师和需要教育资源的学生。精品课程网将主要解决两个群体之间的资源共享的问题，在教师与学生之间充当资源共享的平台，在确立了这一目标后，我们可以针对两者的需求确定出不同的功能，以满足双方的需求。

#### 1.1学生需求分析

1. 可以登录网站浏览和查找各种信息以及下载或在线浏览文件。
2. 可进行完整的页面访问与用户自测。学生是课程系统的使用者兼评价者。
3. 可参与视频、音频、图文形式的直播课堂；进行内容付费，如购买课程。
4. 可会回看直播课程；可以进行在线随堂测试；可通过文字互动或者语言视频的形式，与讲师进行实时讨论；可浏览教程，教师信息；可对课程做出评价。

#### 1.2教师需求分析

1. 可以登录网站浏览、上传、发布、修改、更新自身所有的教学资源。
2. 可对自身所有的教学资源进行收集和管理。
3. 可以进行视频、音频、图文形式的直播授课；可在课堂场景中与学生进行文字或语言互动。可发布随堂测试或作业等。

### 2．性能需求分析

平台的正常运行，离不开硬件和软件环境的支持，由于需要引入直播形式的教学，必须保证网络负载能达到足以承载学生正常观看的情况，由于系统的使用人数未设置上限，该系统按10人同时在线观看的承载能力进行设计，需保证10人同时观看时不出现因系统原因造成的卡顿或断线情况发生。在其余界面访问时，需保证100人同时访问该系统不会因系统原因出现报错或卡顿情况发生，实际量化为在500ms以内能够正常访问此系统。最终系统测试将以此为基准做出系统评价。

（二）系统开发环境

1．Vue.js简介

Vue是一套用于构建用户界面的渐进式JavaScript框架。与其它大型框架不同的是，Vue被设计为可以自底向上逐层应用。Vue的核心只关注视图层，方便与第三方库或既有项目整合。另一方面，Vue完全有能力驱动采用单文组件和Vue生态系统支持的开发的复杂单页应用。Vue.js的目标是通过尽可能简单的API实现响应的数据绑定和组合视图组件。Vue.js自身不是一个全能的框架，它只聚焦于视图层，因此它非常容易学习，非常容易与其它库或已有项目整合，在与相关工具和支持库一起使用时，Vue.js也能完美地驱动复杂的单页应用。其运行大小仅为20kb min+gzip，拥有超快速的DOM响应和最少的优化，足以应付任何规模的应用。

2．SpringBoot简介

SpringBoot是由Pivotal团队研发的开源版本的轻量级框架，它基于Spring4.0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。

3．ElasticSearch简介

ElasticSearch是一个分布式、高扩展、高实时的搜索与数据分析引擎，基于RESTful web接口，使用Java语言进行开发，并作为Apache许可条款下开放源码，是一种流行的企业级搜索引擎。具有实时、稳定、可靠、快速、安装使用方便的特性。

4．RabbitMQ简介

RabbitMQ是实现了高级消息队列协议（AMQP）的开源消息代理软件。RabbitMQ服务器是用Erlang语言编写的，可以快速的解决异构系统的不同进程间相互调用、数据交换问题。具有可伸缩性、消息持久化性、稳定、快速的特点。

5．Redis简介

Redis（Remote Dictionary Server），即远程字典服务，是一个开源的使用ANSO C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Vaule数据库，并提供多种语言的API。

6．MySql简介

MySql是一款安全、跨平台、高效的，并与PHP、Java等主流编程语言紧密结合的数据库系统。由于其体积小、速度快、总体拥有成本低，尤其是源码开放的特点，使得很多公司都采用MySql数据库以降低成本。

7．Nginx简介

Nginx是一个高性能的HTTP和反向代理服务器，也是一款轻量级Web服务器及电子邮件代理服务器，在BSD-like协议下开源，具有内存占用少，并发能力强等特点。

8．Docker简介

Docker是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖到一个可移植的容器中，然后发布到任何流行的服务器主机中，可以实现虚拟化，具有自动化部署、轻量、可扩展等特点。

9．Java语言简介

Java语言是在二十世纪末由Sun公司发布的，而且公开源代码，这一优点吸引了许多世界各地优秀的编程爱好者，也使得他们开发出当时一款又一款经典好玩的小游戏。Java语言是纯面向对象语言之一，从发布初期到现今，可以说有将近20多年的历史，已发展成为人类计算机编程语言发展史上的一个深远影响。

Java语言自从正式发布以后。在现在的编程当中已经成为了主要的应用语言。自从进入网络时代以后，在J2EE的应用上它得到了很好的发展和应用，它的一些线性和了扩展性都成为了一个个开发商的应用对象。对于一些就Java语言的当中的安全、跨平台、安全等等都在网络当中得到了很大的应用，它的一些应用和优越性，可以让许多的编程者在时间上缩短了许多和更加方便。

Java语言具有非常多种的特性：（1）跨平台的无关性；（2）面向对象； （3）安全性得以保障；（4）支持多个任务；（5）多种编写方式，代码编写简单。对比其他的低级语言、高级语言，Java语言具有明显的显著优势以及未来开阔的前景，可以广泛的应用在个人笔记本电脑、大数据、大型游戏等等。

Java语言具有面向对象的特性，并且易于理解。关于对象，其实可以理解成每一种事物都是一种对象，包括我们人类自身都是一种对象。利用面向对象语言的基本特征来解决软件开发中的实际问题，为有效软件开发提供了技术支持。

Java 语言具有很好的跨平台无关性。所编写出来的应用程序是Java语言编写的，那么就无需再使用编译器来修改程序代码，可以直接在任何计算机系统中运行，Windows系统可以运行，在Linux系统中也可以，也就是经过一次编译，可以到处运行，所以Java语言具有卓越的可移植性，可以很好的跨平台实现。

# 四、系统总体设计

## （一）系统用例分析

### 1.学生用户用例图

学生作为系统的使用者，可通过系统进行登录，对自己的账号信息进行管理，以及使用和浏览系统中的信息等，如图4.1.1。

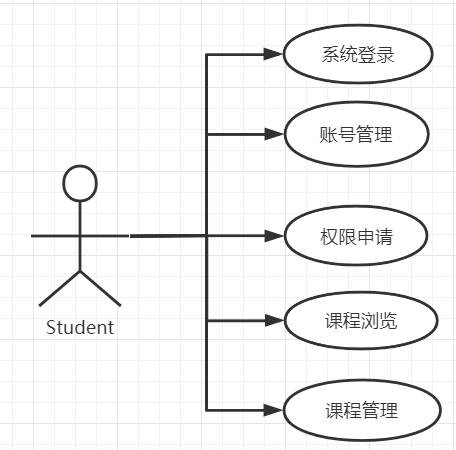


图4.1.1

### 2.教师用户用例图

教师可通过系统进行登录、课程编排、账号信息修改、以及浏览系统信息等，如图4.1.2。

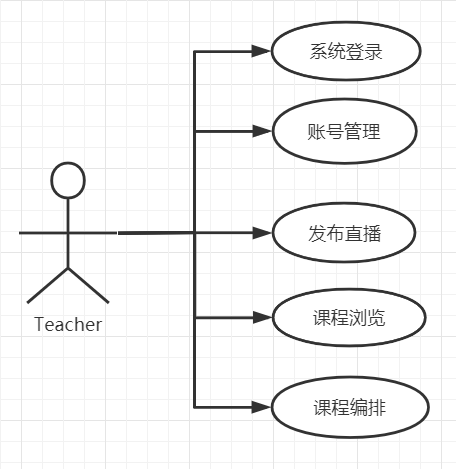


图4.1.2

### 3.管理员用户用例图

管理员可通过系统进行登录、用户管理、权限管理、系统内容管理等，如图4.1.3。

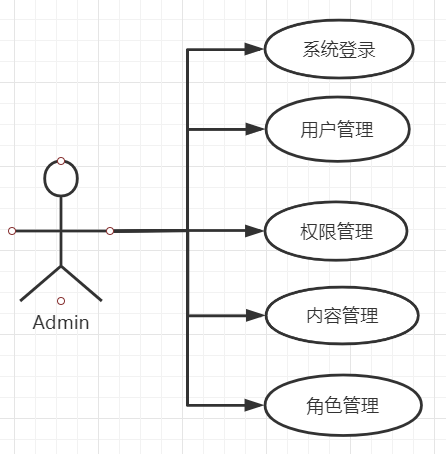


图4.1.3

## （二）系统架构

### 1.系统概述

进过系统用例分析后，就开始记性系统的设计，系统设计包含总体设计和详细设计。总体设计只是一个大体的设计，经过了总体设计，我们能够划分出系统的一些东西，例如文件、文档、数据等。而且我们通过总体设计，大致可以划分出了程序的模块，以及功能。但是只是一个初步的分类，并没有真正的实现。

整体设计，只是一个初步设计，而且，对于一个项目，我们可以进行多个整体设计，通过对比，包括性能的对比、成本的对比、效益的对比，来最终确定一个最优的设计方案，选择优秀的整体设计可以降低开发成本，增加公司效益，从这一点来讲，整体设计还是非常重要的。

精品课程网工作原理图如图4.2.1.1。



图4.2.1.1

精品课程网系统架构图如图4.2.1.2。

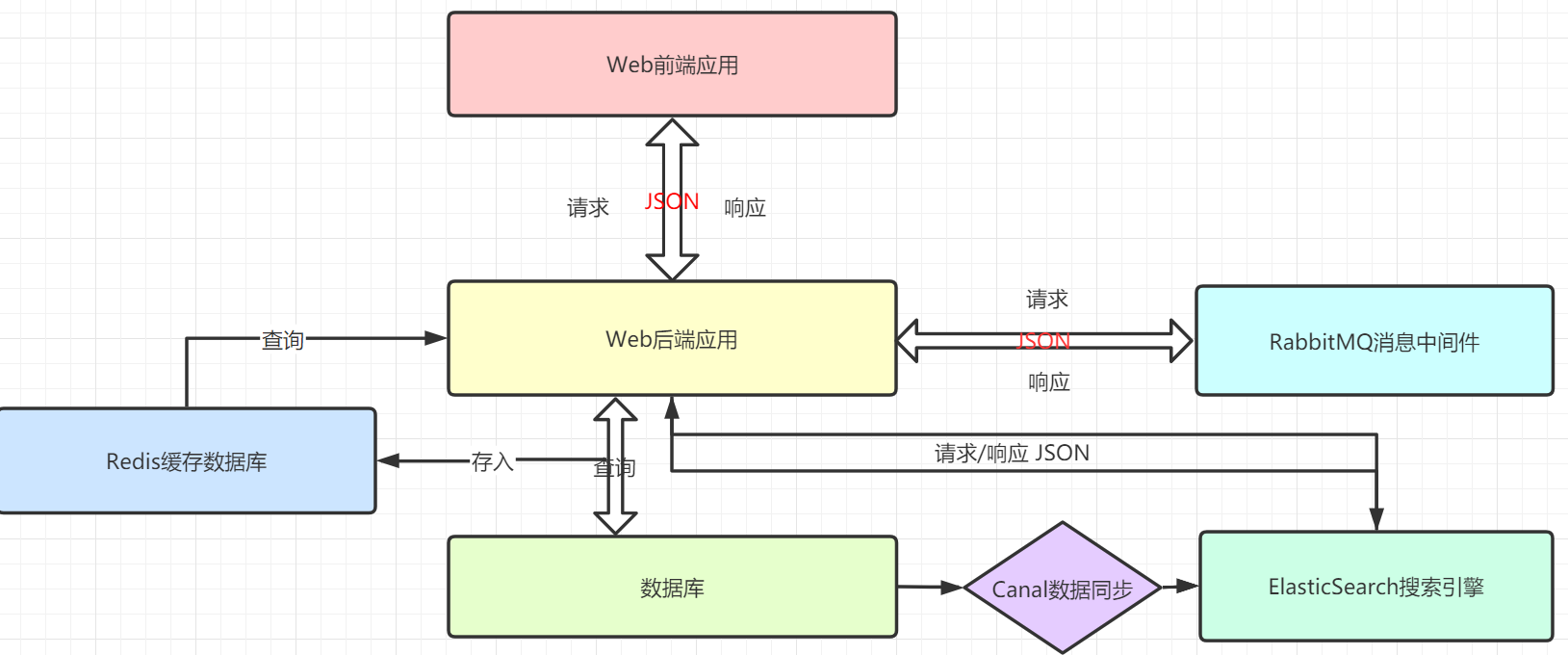


图4.2.1.2

### 2.系统结构设计

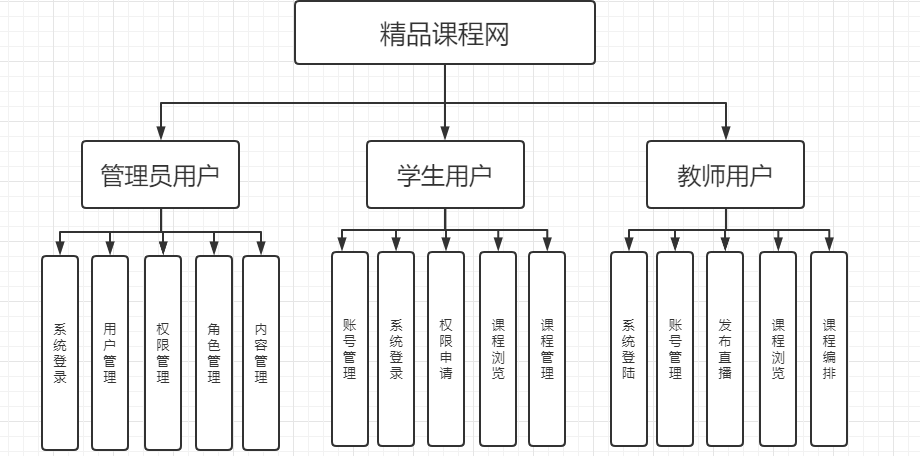
系统架构图属于系统设计阶段，系统架构图只是这个阶段一个产物，系统的总体架构决定了整个系统的模式，是系统的基础。高校网上教材征订系统的整体结构设计如图4.2.2.1所示。

图4.2.2.1

## （三）系统流程

### 1.注册流程

根据国家公布的《互联网跟帖评论服务关联规定》，要求网站主题需要实名制，如果注册用户没有提供相匹配的身份信息认真，网站将不能向其提供更帖评论的服务。精品课程网是面向学生、教师、及其它社会群体开放的教育教学平台，向用户提供了评论的服务，为满足系统的法律可行性，在用户注册系统时，需要提供相应的身份信息进行认证如：个人手机号。通过发送验证码的形式进行对用户信息的唯一绑定。为防止网络攻击，随意发送验证信息的行为，必须加以逻辑上的控制。具体流程如图4.3.1.1和图4.3.1.2。

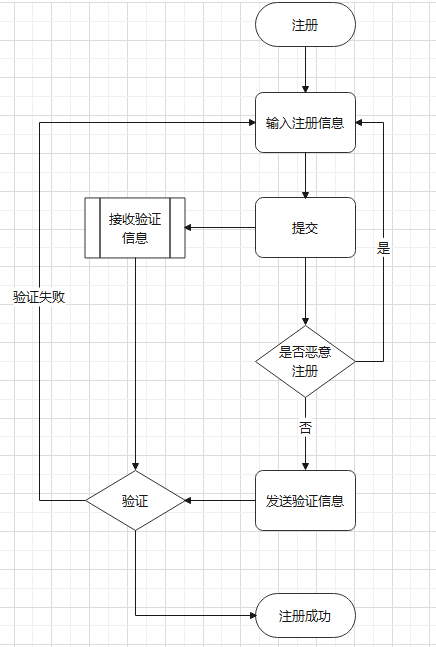


图4.3.1.1

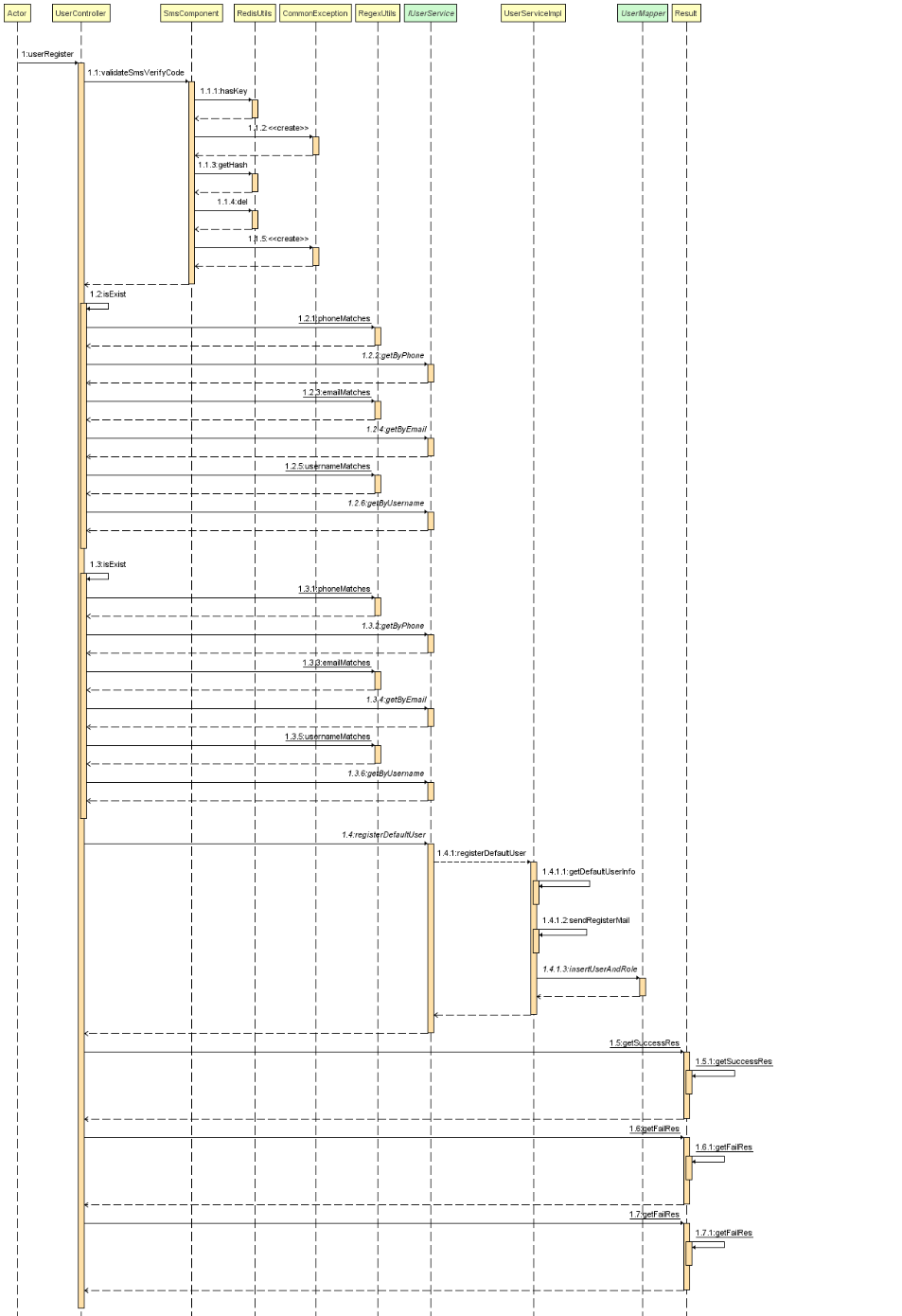


图4.3.1.2注册时序图

### 2.登录流程

为了保证系统的安全性，要使用本系统对系统信息进行管理，必须先登陆到系统中。如图4.3.2.1和图4.3.2.2。

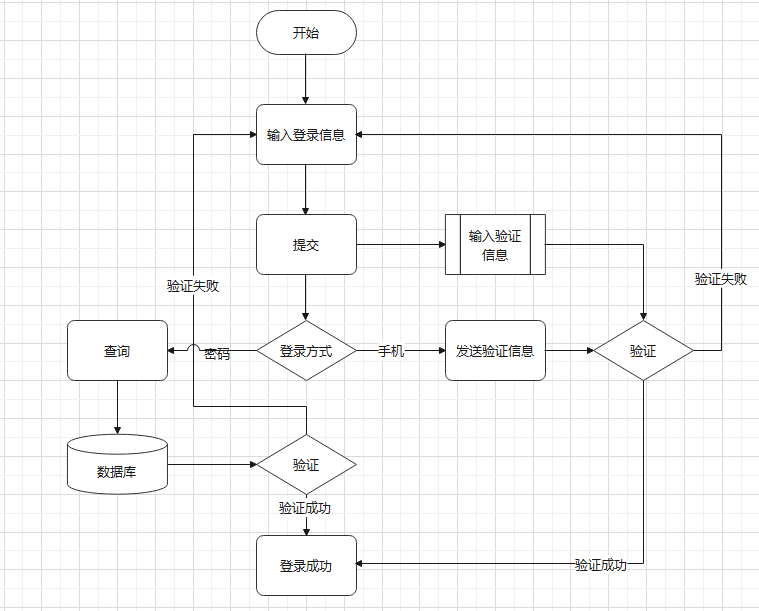


图4.3.2.1 登录流程图

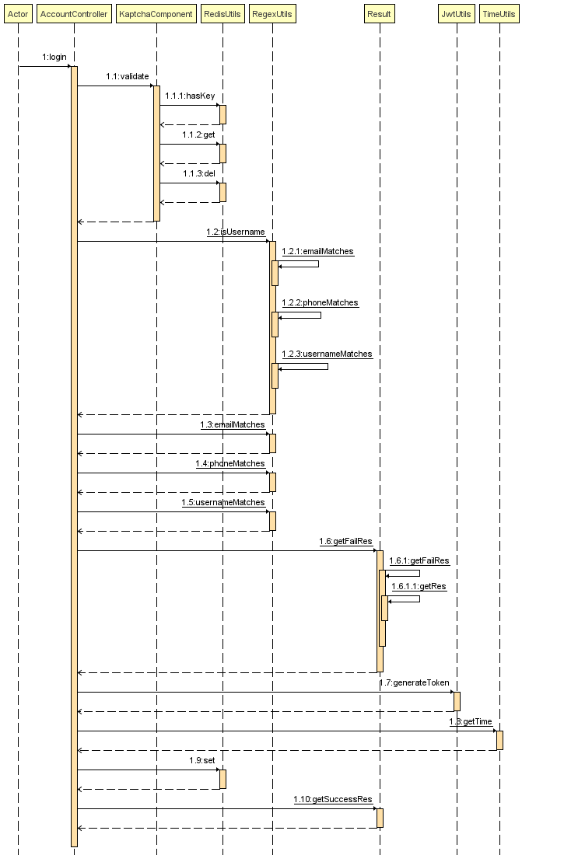


图4.3.2.2 登录时序图

### 3.修改信息流程

管理员可以对教材信息等进行信息的修改，教师也可以对自己权限内的信息进行修改，首先进入修改信息界面，输入修改信息数据，系统进行数据的判断验证，修改信息合法则修改成功，信息更新至数据库，信息不合法则修改失败，重新输入。修改信息流程图如图4.3.3.1所示。

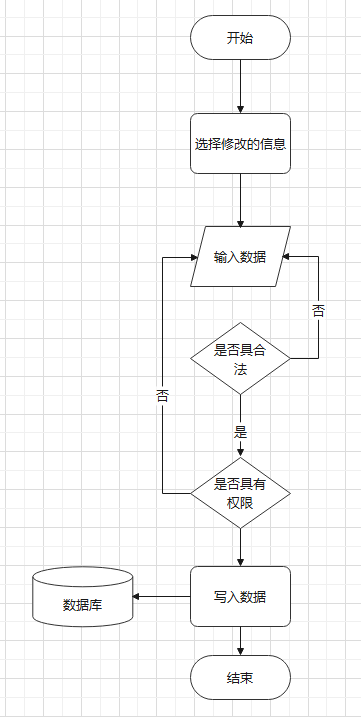


图4.3.3.1

## （四）数据库设计

数据库是计算机信息系统的基础。目前，电脑系统的关键与核心部分就是数据库。数据库开发的优劣对整个系统的质量和速度有着直接影响。

### 1. 数据库设计原则

概念模式它主要是建立在数据需求分析的基础上，它通常是用概念数据模型来表示各个数据之间的联系，并且对系统用户进行信息的处理和管理，同时建立起E-R图来表示具体的实体、属性和联系的关系。

### 2.数据库实体

数据模型中的实体（Entity），也称为实例，对应现实世界中可区别于其他对象的“事件”或“事物”。例如，公司中的每个员工，家里中的每个家具。

本系统的E-R图如下所示：

(1)用户实体图如图4.4.2.1

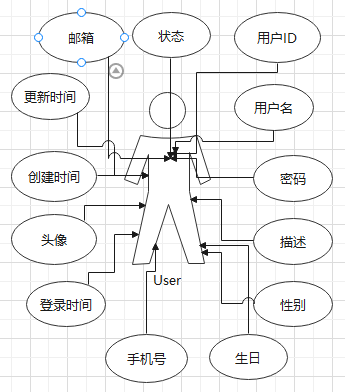


图4.4.2.1

(2)教师实体图如图4.3.2.2

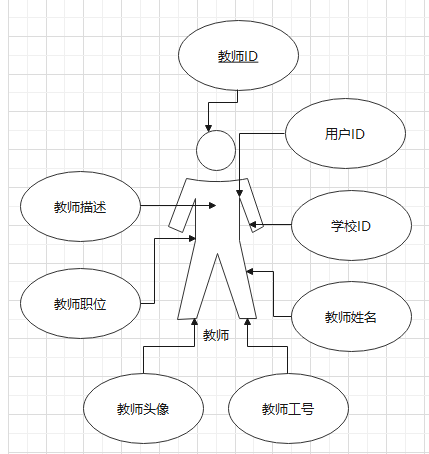


图4.4.2.2

(3)学生实体图如图4.4.2.3

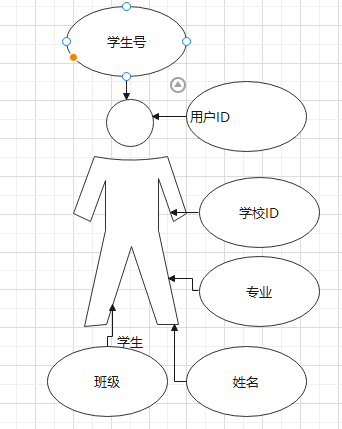


图4.4.2.3

(4)课程实体图如4.4.2.4

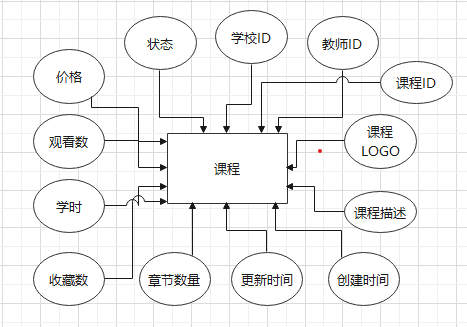


图4.4.2.4

(5)角色实体图如图4.4.2.5

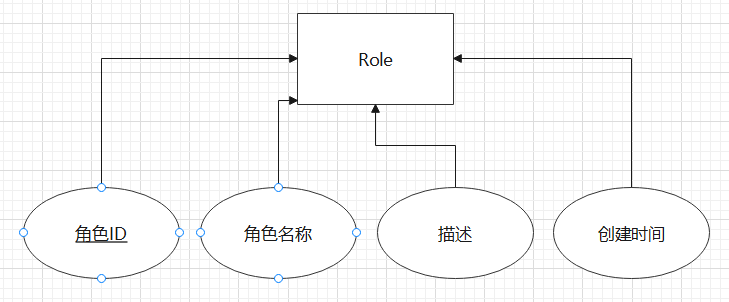


图4.4.2.5

(6)权限实体图如图4.4.2.6

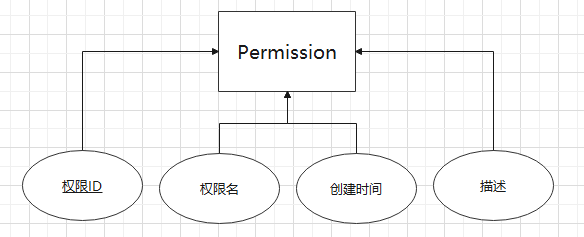


图4.4.2.6

(7)学校实体图如图4.4.2.7

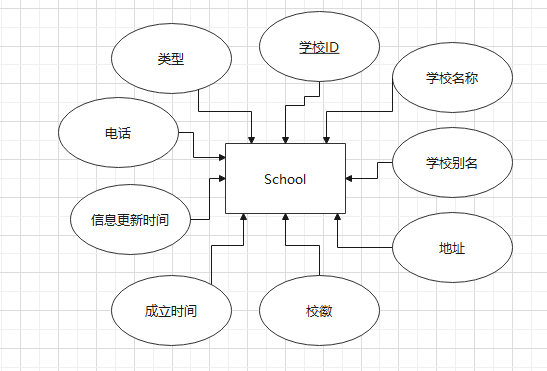


图4.4.2.7

(8)专业实体图如图4.4.2.8

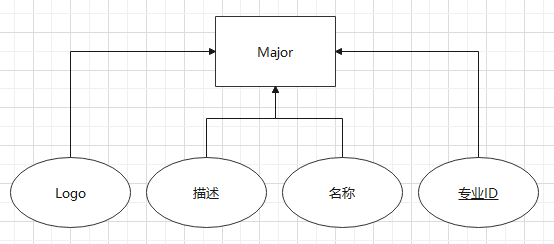


图4.4.2.8

(9)直播课程实体图如图4.4.2.9

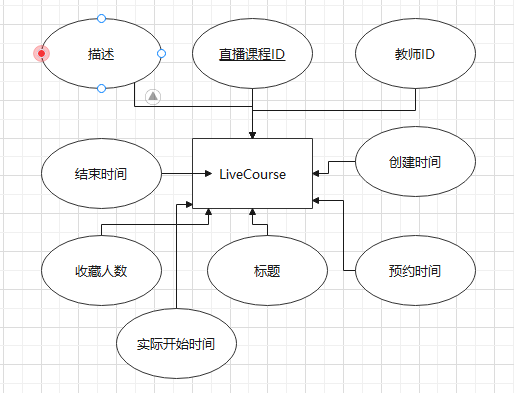


图4.4.2.9

(10)课程章节实体图如图4.4.2.10

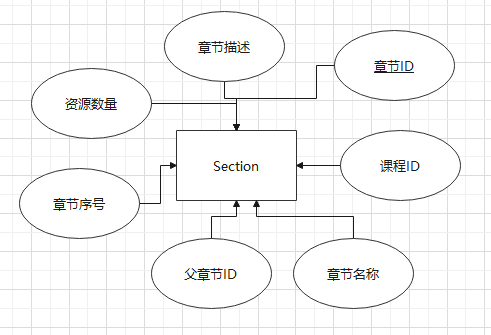


图4.4.2.10

(11)资源实体图如图4.4.2.11

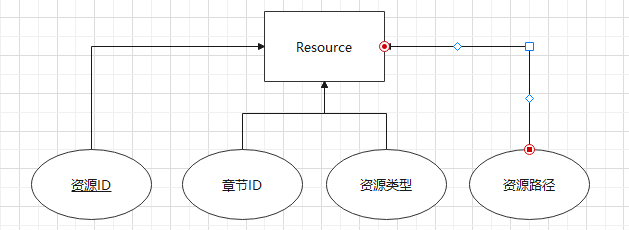


图4.4.2.11

(12)标签实体图如图4.4.2.12

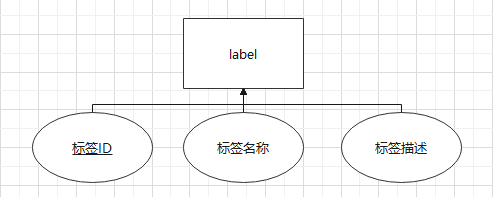


图4.4.2.12

(13)类别实体图如图4.4.2.13

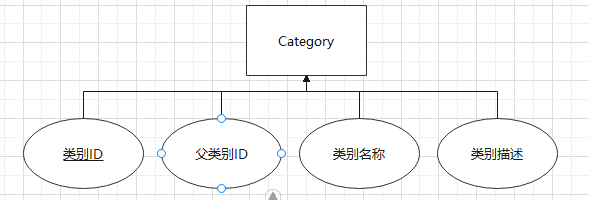


图4.4.2.13

(14)评论实体图如图4.4.2.14

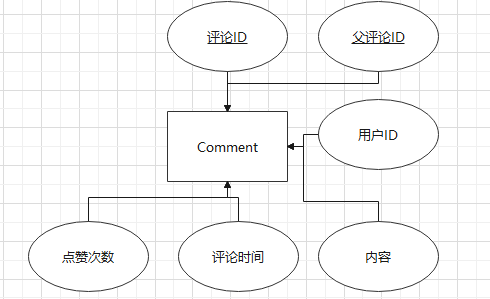


图4.4.2.14

### 3.数据库表设计

(1)用户表，用于存放用户的基本信息包括密码，密码通过MD5散列加密加盐获得密文形式存储。

表4.4.3.1 t\_user用户表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| user\_id | varchar(32) primary key | 用户id，主键，无实际意义，uuid |
| username | varchar(16) not null unique | 用户名，最大长度4-16字符，不可重复 |
| password | varchar(16) not null | 用户密码，16字符 |
| user\_desc | varchar(128) | 用户描述 |
| user\_sex | int not null default 0 | 用户性别{0：保密，1：男，2：女}，默认为0 |
| user\_birthday | date() | 用户出生日期 |
| user\_phone | varchar(11) | 手机号码，11位数字 |
| user\_email | varchar(32) not null | 电子邮箱 |
| user\_avatar | varcahr(256) | 用户头像 |
| user\_created | datetime() not null default now() | 账户创建时间 |
| user\_login | datetime() | 上次登录时间 |
| user\_update | datetime not null default now() | 用户信息更新时间 |
| user\_status | int default 0 | 用户状态，0已激活，1未激活，2邮箱激活，3电话激活。 |

（2）学校表，用于存放学校信息

表4.4.3.2 t\_school学校表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| user\_id | varchar(32) pri表4.3.3.2 t\_school学校表  mary key | 用户id，主键，无实际意义，uuid |
| username | varchar(16) not null unique | 用户名，最大长度4-16字符，不可重复 |
| password | varchar(16) not null | 用户密码，16字符 |
| user\_desc | varchar(128) | 用户描述 |
| user\_sex | int not null default 0 | 用户性别{0：保密，1：男，2：女}，默认为0 |
| user\_birthday | date() | 用户出生日期 |
| user\_phone | varchar(11) | 手机号码，11位数字 |
| user\_email | varchar(32) not null | 电子邮箱 |
| user\_avatar | varcahr(256) | 用户头像 |
| user\_created | datetime() not null default now() | 账户创建时间 |
| user\_login | datetime() | 上次登录时间 |
| user\_update | datetime not null default now() | 用户信息更新时间 |
| user\_status | int default 0 | 用户状态，0已激活，1未激活，2邮箱激活，3电话激活。 |

（3）专业表，用于存储学校的专业信息

表4.4.3.3 t\_major专业表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| inner\_id | varchar(32) primary key | 中间表id，主键，无实际意义，uuid |
| school\_id | varcahr(32) | 学校id，外键 |
| major | varcahr(32) | 专业id，外键 |

（4）班级表，用于存储班级信息

表4.4.3.4 t\_class班级表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| class\_id | varchar(32) primary key | 班级id，主键，无实际意义，uuid |
| school\_id | varchar(32) not null | 学校id，与学校绑定 |
| major\_id | varchar(32) not null | 专业id，与专业绑定 |
| class\_no | varchar(32) not null | 班级号 |
| class\_grade | date() not null | 班级年级 |

（5）学生表，用于存储学生信息

表4.4.3.5 t\_student 学生表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| student\_id | varchar(32) primary key | 学生id，主键，无实际意义，uuid |
| user\_id | varchar(32) not null | 用户id，与用户账号绑定 |
| school\_id | varchar(32) not null | 学校id，与学校绑定 |
| major\_id | varchar(32) not null | 专业id，与专业绑定 |
| class\_id | varchar(32) not null | 班级id，与班级绑定 |
| student\_no | varchar(64) not null | 学生号 |
| student\_name | varcahr(16) not null | 学生姓名，最大长度16字符 |

（6）教师表，用于存储教师信息

表4.4.3.6 t\_teacher教师表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| teacher\_id | varcahr(32) primary key | 教师id，主键，无实际意义，uuid |
| user\_id | varchar(32) not null | 用户id，与用户账号绑定 |
| school\_id | varchar(32) not null | 学校id，与学校绑定 |
| teacher\_no | varchar(32) not null | 教师号 |
| teacher\_name | varchar(16) not null | 教师姓名，最大长度16字符 |
| teacher\_avatar | varhcar(256) | 头像 |
| teacher\_desc | varchar(256) | 描述 |
| teacher\_post | varhcar(10) | 教师职位 |

（7）课程表，用于存储课程信息

表4.4.3.7 t\_course课程表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| course\_id | varchar(32) primary key | 课程id，主键，无实际意义，uuid |
| teacher\_id | varchar(32) | 教师id |
| course\_name | varchar(16) not null | 课程名，最大长度16字符 |
| course\_desc | varchar(256) not null | 课程描述，最大长度256字符 |
| course\_logo | varchar(128) | 课程Logo |
| course\_created | datetime() not null defaut now() | 课程创建日期 |
| course\_updated | datetime() not nulldefault now() | 课程更新日期 |
| section\_count | int not null | 课程章节数量，通过设置计算 |
| course\_hour | int not null | 课程学时，必须手动设置，不能为0 |
| course\_views | long not null default 0 | 课程观看次数，默认为0 |
| course\_star | long not null default 0 | 课程收藏次数，默认为0 |
| course\_price | decimal(5,2) not null default 0.00 | 课程价格，默认为0 |
| course\_status | int not null default 0 | 课程状态，{0：开放，1：关闭}，默认为0 |

（8）课程章节表，用于存储课程章节信息

表4.4.3.8 t\_course\_section 课程章节表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段名称 |
| section\_id | varchar(32) primary key | 章节id，主键，无实际意义，uuid |
| course\_id | varchar(32) not null | 课程id，与课程绑定 |
| section\_name | varchar(16) not null | 章节名称，最大长度16字符，不能为空 |
| parent\_id | varchar(32) not null default 0 | 父章节id，默认为0，表示当前为根目录 |
| section\_no | int not null | 章节序号，查询时排序使用 |
| resource\_count | int not null | 当前资源数量，通过资源计算 |
| section\_desc | varchar(128) | 章节描述，最大长度128字符 |

（9）课程资源表，用于存放每个章节的资源

表4.4.3.9 t\_course\_resource

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| resource\_id | varchar(32) primary key | 资源id，主键，无实际意义，uuid |
| section\_id | varchar(32) not null | 章节id，与章节绑定 |
| resource\_type | int not null | 资源类型{0:图片，1:视频，2:PPT，3:word, 4:其它} |
| resource\_path | varchar(256) not null | 资源路径地址，上传后自动获取 |

（10）班级和课程中间表，将课程和班级相关联

表4.4.3.10 tm\_class\_course 班级课程中间表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| inner\_id | varchar(32) primary key | 中间表id，主键，无实际意义，uuid |
| class\_id | varchar(32) not null | 班级id，绑定班级 |
| class\_type | int not null | 班级类型：{0：系统开设，1：院校开设} |
| course\_id | varchar(32) not null | 课程id，绑定课程 |
| teacher\_id | varchar(32) not null | 开设老师，绑定教师表，{系统开设默认为课程老师} |
| create\_time | date() | 开设日期 |
| finish\_time | date() | 结束日期 |
| credit | double not null default 0.0 | 学分，{系统开设默认为0} |
| schedule | double not null default 0.0 | 班级学习进度，默认为0，通过班级平均进度计算 |

（11）用户和课程中间表，将用户与课程相关联

表4.4.3.11 tm\_user\_course 用户课程中间表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| inner\_id | varchar(32) primary key | 中间表id，主键，无实际意义，uuid |
| user\_id | varchar(32) not null | 用户id，与用户绑定 |
| course\_id | varchar(32) not null | 课程id，与课程绑定 |

（12）标签表，存储课程标签信息

表4.4.3.12 t\_label 标签表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| label\_id | varchar(32) primary key | 标签id，主键，无意义，uuid |
| label\_name | varchar(16) not null unique | 标签名称，最大长度16字符，不能重复 |
| label\_desc | varchar(128) | 标签描述 |

（13）课程和标签中间表，将课程和标签相关联

表4.4.3.13 tm\_course\_label 课程标签中间表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| inner\_id | varchar(32) primary key | 中间表id，主键，无实际意义，uuid |
| label\_id | varchar(32) not null | 标签id，与标签绑定 |
| course\_id | varchar(32) not null | 课程id，与课程绑定 |

（14）类别表，存储课程类别信息。

表4.4.3.14 t\_catgetory 类别表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| category\_id | varchar(32) primary key | 类别id，主键，无实际意义，uuid |
| category\_name | varchar(16) not null unique | 类别名称，不能重复 |
| category\_desc | varchar(128) | 类别描述 |
| parent\_id | varchar(32) not null default 0 | 父类别id，根类别默认为0 |

（15）类别和课程中间表，将类别和课程相关联。

表4.4.3.15 tm\_course\_category 课程类别中间表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| inner\_id | varchar(32) primary key | 中间表id，主键，无实际意义，uuid |
| course\_id | varchar(32) not null | 课程id，绑定课程 |
| category\_id | varchar(32) not null | 类别id，绑定类别 |

（16）评论表，存储课程相关的评论信息。

表4.4.3.16 t\_mcomment 评论表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| comment\_id | varchar(32) primary key | 评论id，主键，无实际意义，uuid |
| course\_id | varchar(32) not null | 课程id，与课程绑定 |
| user\_id | varchar(32) not null | 用户id，与用户绑定 |
| parent\_id | varchar(32) not null default 0 | 父评论id，如为根评论默认为0 |
| comment\_content | varchar(128) not null | 评论内容，最大长度128个字符，不能为空 |
| comment\_star | int default 0 | 点赞次数，默认为0 |

（17）角色表，存储权限相关的角色信息。

表4.4.3.17 t\_role 角色表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| role\_id | varchar(32) primary key | 角色id，主键，无实际意义，uuid |
| role\_name | varchar(16) not null | 角色名称 |
| role\_created | datetime() not null default now() | 创建时间 |
| role\_desc | varchar(256) | 角色描述 |

（18）用户角色中间表，将角色与用户相关联。

表4.4.3.18 tm\_user\_role

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| inner\_id | varchar(32) primary key | 中间表id，主键，无实际意义，uuid |
| user\_id | varchar(32) | 用户id，与用户绑定 |
| role\_id | varchar(32) | 角色id，与角色绑定 |
| created | datetime not null default now() | 创建时间 |

（19）权限表，存储系统权限信息。

表4.4.3.19 t\_permission 权限表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| permission\_id | varchar(32) primary key | 权限表id，主键，无实际意义，uuid |
| permission\_name | varchar(16) not null | 权限名称 |
| permission\_created | datetime() default now() | 权限创建时间 |
| permission\_desc | varchar(128)` | 权限描述 |

（20）角色权限中间表，将权限信息赋予角色。

表4.4.3.20 tm\_ role\_permission 角色权限中间表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| inner\_id | varchar(32) primary key | 中间表id，主键，无实际意义，uuid |
| role\_id | varchar(32) | 角色id，与角色绑定 |
| permission\_id | varchar(32) | 权限id，与权限绑定 |
| created | datetime not null default now() | 创建时间 |

（21）网站布局表，存储网站布局信息。

表4.4.3.21 t\_website\_layout 网站布局表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| layout\_id | varhcar(32) primary key | 布局id，主键，无实际意义,uuid |
| layout\_name | varchar(32) not null | 布局名称 |
| layout\_type | int not null | 布局类型，{0：文字，1：图片，2：视频，3：音频} |
| layout\_page | varchar(32) not null | 布局页面，如：首页 |
| layout\_location | varchar(32) not null | 布局位置，如：导航栏 |
| layout\_desc | varchar(128) | 布局描述 |

（22）直播课程表，用于存储直播信息。

表4.4.3.22 t\_live\_course 直播课程表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| live*course*id | varchar(32) primary key | id |
| teacher\_id | varchar(32) | 教师id |
| created | date | 创建时间 |
| reserve\_time | date | 预定直播时间 |
| title | varchar(32) | 直播标题 |
| star | int | 预约人数 |
| finished | date | 结束时间 |
| live*course*desc | varchar(256) | 直播课程描述 |
| start | date | 开始时间，直播时记录 |

（23）课程章节评论表，用于存储章节的评论信息。

表4.4.3.23 t\_section\_comment 章节评论表

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 字段描述 |
| comment\_id | varchar(32) | id |
| section\_id | varchar(32) | 章节id |
| user\_id | varchar(32) | 用户id |
| parent\_id | varchar(32) | 父评论id，根评论为0 |
| comment\_content | varchar(256) | 评论内容 |
| comment\_star | int default 0 | 点赞次数 |
| comment\_time | datetime | 评论时间 |

# 五、系统详细设计

## （一）界面设计

### 1.登录页面

用户可以选择多种方式进行登录，并且需要输入验证码进行校验，如果校验失败会给出相应的提示，如果我们输入的账号、密码不正确，数据库就会提示出错误的信息提示，同时会提示用户重新输入自己的账号、密码，直到账号密码输入成功后，会提登录成功的信息，并返回网站首页。如图5.1.1.1和图5.1.1.2。



图5.1.1.1



图5.1.1.2