

毕 业 论 文

题 目 计算机组成原理培训学习平台

学 院 计算机科学与软件学院

专 业 计算机科学与技术

年 级 2019级

学 号 201924111303

学生姓名 卢津瑜

指导教师 王俊波

完成时间 2023 年 4 月

肇庆学院教务处制

**学术诚信声明**

本人所呈交的毕业论文，是在指导教师的指导下独立完成。研究工作所取得的成果、数据、图片资料均真实可靠。除文中已注明引用的内容外，不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究做出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名（手写）： 日期： 年 月 日

**目 录**

摘要与关键词………………………………………………………………………………1

Abstract and Keywords……………………………………………………………………1

[1 绪论 1](#_Toc131029408)

[1.1 课题背景 1](#_Toc131029409)

[1.2 课题意义 2](#_Toc131029410)

[1.3 研究目的 2](#_Toc131029411)

[1.4 国内外研究现状 2](#_Toc131029412)

[2 设计思路 3](#_Toc131029413)

[2.1系统框架 3](#_Toc131029414)

[2.2开发工具以及环境 3](#_Toc131029415)

[3 系统设计 3](#_Toc131029416)

[3.1 体系结构 3](#_Toc131029417)

[3.1.1 用户模块 4](#_Toc131029418)

[3.1.2 安全模块 5](#_Toc131029419)

[3.1.3 邮件模块 6](#_Toc131029420)

[3.1.4 文件上传模块 6](#_Toc131029421)

[3.1.5 习题组模块 7](#_Toc131029422)

[3.1.6 选择题模块 8](#_Toc131029423)

[3.1.7 判断题模块 8](#_Toc131029424)

[3.1.8 主观题模块 9](#_Toc131029425)

[3.1.9 视频模块 9](#_Toc131029426)

[3.1.10 视频历史记录模块 10](#_Toc131029427)

[3.1.11 个人学习进度模块 11](#_Toc131029428)

[4 数据库设计 11](#_Toc131029429)

[4.1 用户表 13](#_Toc131029430)

[4.2 角色表 13](#_Toc131029431)

[4.3 用户角色表 14](#_Toc131029432)

[4.4 视频表 14](#_Toc131029433)

[4.5 视频历史记录时间表 15](#_Toc131029434)

[4.6 视频播放量表 15](#_Toc131029435)

[4.7 习题组表 16](#_Toc131029436)

[4.8 选择题表 17](#_Toc131029437)

[4.9 判断题表 17](#_Toc131029438)

[4.10 主观题表 18](#_Toc131029439)

[4.11 个人学习进度表 18](#_Toc131029440)

[5 系统实现 18](#_Toc131029441)

[5.1 登录实现 18](#_Toc131029442)

[5.2 注册实现 20](#_Toc131029443)

[5.3 视频列表 21](#_Toc131029444)

[5.4 视频播放页面 22](#_Toc131029445)

[5.5 习题总览界面 23](#_Toc131029446)

[5.6 习题界面 23](#_Toc131029447)

[5.7 视频上传界面 24](#_Toc131029448)

[5.8 添加习题 26](#_Toc131029449)

[5.9 首页 27](#_Toc131029450)

[5.10 个人主页 29](#_Toc131029451)

[5.11 管理员界面 31](#_Toc131029452)

[6 结论 33](#_Toc131029453)

参考文献……………………………………………………………………………………2

致 谢………………………………………………………………………………………2

计算机组成原理培训学习平台的设计实现

摘 要**:** 计算机组成原理是计算机专业的核心课程，课程中许多的知识点抽象而且难以理解，并且大多都具有较高的技术性，课程难度较大。且在整个课程体系中占有重要地位。而计算机组成原理培训学习平台是能够提供给学生线上做好课前预习和演练，具有视频学习和习题演练功能，对于复杂的课程内容可以做到及时消化和吸收，也方便教师跟踪和反馈。本系统采用SpringBoot和JQuery技术，还有一下特点：（1）使用MongoDB数据库，能处理高并发的读写和访问请求。（2）对于视频等大文件的上传，采用了文件分片上传、大文件秒传、断点续传的技术，使得上传文件更加安全且快速。

关键词:计算机组成原理; SpringBoot; 分片上传;JQuery

**Abstract**: The principle of computer composition is the core course of computer major, many knowledge points in the course are abstract and difficult to understand, and most of them are of high technology, the course is more difficult. And plays an important role in the whole curriculum system. The computer composition principle training and learning platform can provide students with online pre-class preview and drill, with video learning and exercise drill functions, which can timely digest and absorb complex course content and facilitate teachers to track and feedback. This system adopts SpringBoot and JQuery technologies, and has the following features: (1) It uses MongoDB database, which can handle highly concurrent read and write and access requests. (2) For uploading large files such as videos, it adopts the technologies of file fragment uploading, second uploading of large files, and resumable uploading of breakpoint, making file uploading safer and faster.

**Keyword**: principles of computer composition; SpringBoot; JQuery; Fragment upload

# 1 绪论

## 1.1 课题背景

现在的信息技术不断变化升级，越来越丰富的网站以及应用使得用户更多地愿意接受在线培训平台的学习。在线学习，在线培训一直都是“互联网+”大环境下广受关注的领域之一。计算机组成原理是计算机专业的核心课程，课程中许多的知识点抽象而且难以理解，并且大多都具有较高的技术性，课程难度较大，且在整个课程体系中占有重要地位[7]。采用线下实时教学，学生无法进行较好的课前预习和演练，对于复杂的内容无法做到及时消化和吸收，而且学生对该课程的掌握程度也无法直观的看到，不利于教学的反馈和跟踪，且线下的教学在遇到突发情况时，无法保证正常的教学，严重影响教学进度。

## 1.2 课题意义

在此背景下，本文提出了计算机组成原理课程培训学习平台的设计与实现，本系统就是为了能让学习者更好的学习计算机组成原理这门课程而准备的，平台中有大量的专业视频讲解，使得学习者在空闲时间也可以随时随地的观看视频。为了让学生更好的理解理论知识，训练学生自主思考和动手能力，视频配有一系列的习题供学习者巩固所学知识，这将会让学习者更加充分的深入了解对应课程的内容，使得学习者可以拥有身临其境的感受，提高了学习的积极性[8]。

计算机组成原理课程培训学习平台的设计就是为了解决这样的实际困难。通过该平台，学生可以自主上网站预习或者复习，在课堂上没吸收的知识也可以在课后做到很好的消化，教师也可以根据学生在平台的学习情况及时得到反馈[1]。

## 1.3 研究目的

本研究主要目的在于：

1. 通过计算机组成原理培训平台的网络学习方法，可以拓宽学生们学习计算机组成原理课程的来源，给学生们多一种学习方式，也通过本平台的网络教育，旨在激发学生们对本课程的学习兴趣，提高学生自我学习能力，提高积极性。
2. 打破传统的线下教育，在线上也能很好的掌握课程知识点以及重点难点，学生在线下课堂中遇到较难的知识点但又未吃透，可以课后来到平台中反复观看视频，更好的掌握知识点及考点，学生能在平台上进行课前预习，课后复习，极大的提高了学生们对本课程学习的积极性，也提高了学习的自主性。教师也可以在课后对学生的学习情况进行一定的跟踪和得到一定反馈，课堂上未讲解的知识点也可以利用平台中的视频功能，让学生们在课后观看，对知识反复吸收。可以通过习题给学生们巩固课堂中学习过的知识点。
3. 培养兴趣

## 1.4 国内外研究现状

国内因为疫情的特殊时期，同时伴随着网络技术的发达和自媒体行业的突飞猛进，国内教育学习视频平台也层出不穷，其中有各种各样的类型，有短视频，长视频，互动视频等等方式推广视频学习，通过视频辅助来学习，发挥视频作用[10]。

国外有研究音视频媒体对培养幼儿说话能力，其中有提到孩子在面对这类视频媒体形式的教学时，更能提起孩子的积极性，并且愿意倾向于以这种方式去学习[11]。这也体现了以视频观看为主导的学习方式，能更好的引导学习者带入情景。在线教学的发展，使得在线模式中的在线课程学习成果里，学生学习满意程度不断地提升[12]。

由此可见，对于一个在线的视频学习平台，能够满足学生的学习需求，也能提高学生们的学习的积极性。

# 2 设计思路

## 2.1系统框架

前端部分采用HTML、CSS、JavaScript以及JQuery技术编写再予以Boostrap辅助优化界面美观，JQuery框架的兼容性较强，凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发[9]。并且拥有强大的选择器，轻量级，出色的DOM操作的封装，完善的文档。后端部分采用SpringBoot、MyBatis Plus和SpringSecurity框架，集成了第三方配置，搭建速度和开发更方便。数据存储方面采用MySQL和MongoDB数据库。

## 2.2开发工具以及环境

使用Intellij IDEA 开发工具开发，图形化数据库管理数据软件使用 Navicat Premium 15，使用PostMan和Apifox做接口测试，使用Git、Maven做版本控制管理。

# 3 系统设计

## 3.1 体系结构

如图所示，计算机组成原理培训学习平台的后端部分由以上六大模块为主要组成部分，其中用户模块中包含安全模块。邮件模块包含验证码模块，习题组模块包含选择题、判断题和主观题三大模块，视频模块中包含视频历史记录模块。可以看出，每个模块都有自己的功能，若是想增添功能以及模块，不需要对整个系统结构进行改动，在原先的基础上增添即可。如下图3.1所示。

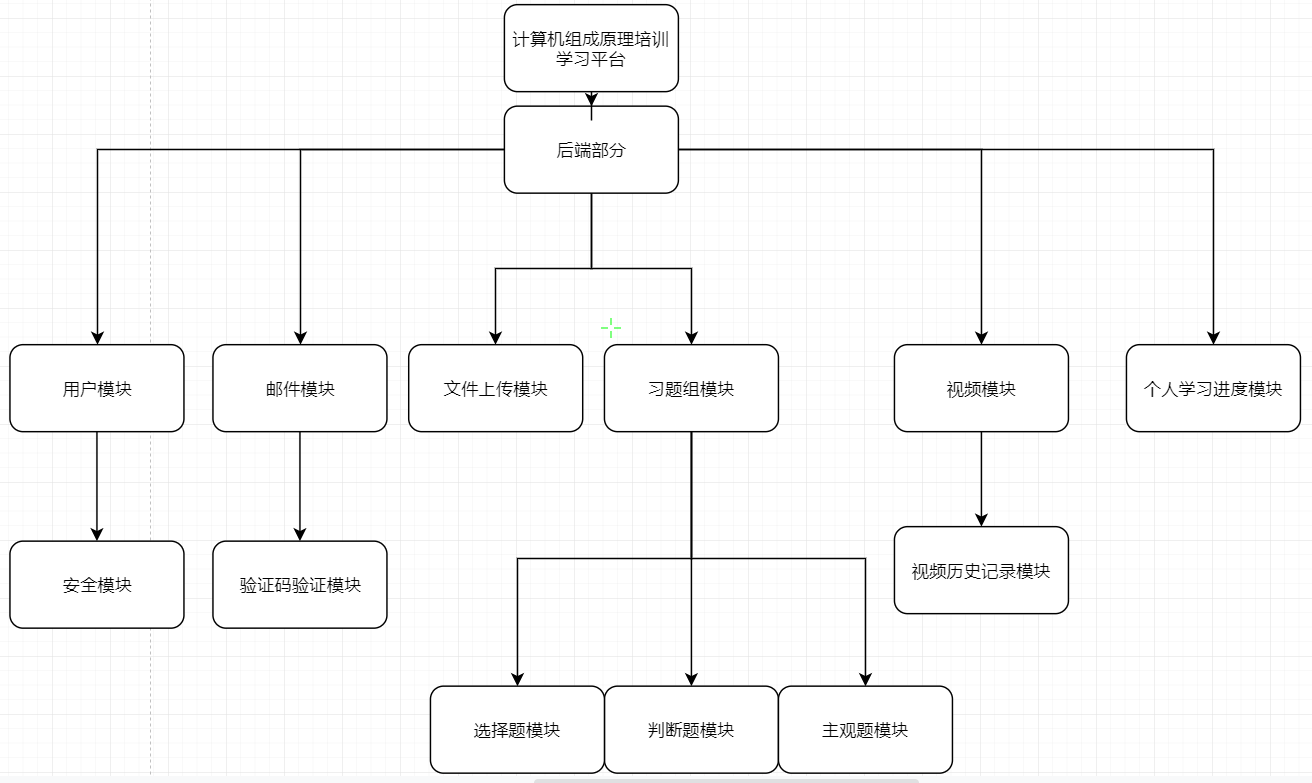


图 3-1

### 3.1.1 用户模块

用户的主要功能登录以及注册，登录校验中会检查用户密码是否正确以及账号是否存在，登录成功之后，用户个人账号信息会暂时存在Redis缓存之中，并且返回一个token给到客户端方便用户后续的访问其他模块操作。用户发送注册请求，服务器接受注册信息，经过验证模块、昵称查重检测和邮箱查重检测之后，用户注册成功，并且将用户信息保存到数据库用户表中。登录功能核心代码实现详见附录A1。模块数据流图如下图3-2所示

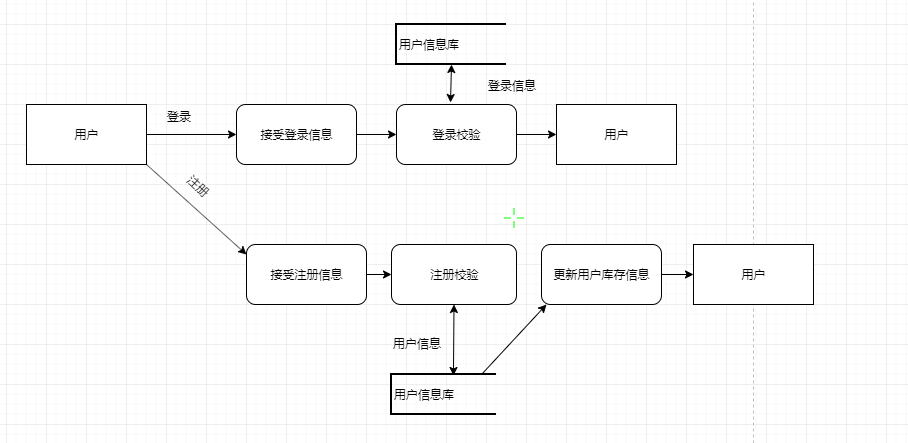


图 3‑2

### 3.1.2 安全模块

安全模块主要处理所有请求包括用户的访问请求，如果是已经登录的用户，会携带一个具有过期时间的token，访问过滤器将会拦截本次访问，并且如果携带token，将会经过处理和与服务器中的缓存做对比，得到用户基本信息，在查询该用户所拥有的权限，将合并之后的信息提供给服务端。遇到了跨域请求时，跨域过滤器将会对本次请求做一定处理，将结果由服务端发送回客户端。模块数据流图如下图3-3所示。

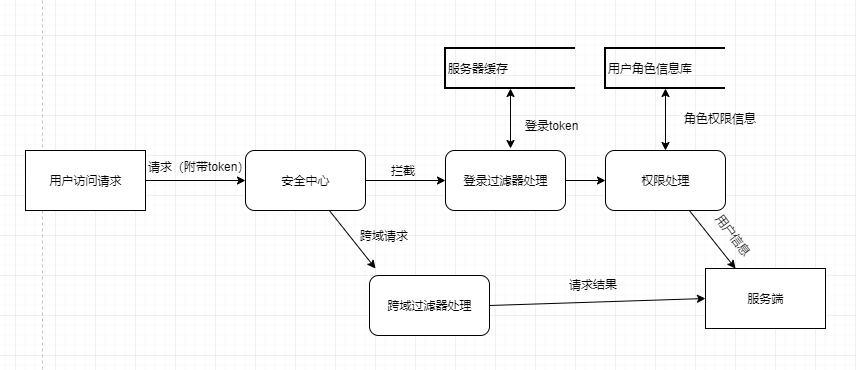


图 3-3

### 3.1.3 邮件模块

邮件模块主要分为三大块，首先是在接收了验证码请求之后，接收来自客户端的邮箱号码信息，对邮箱号码进行基本验证，通过验证之后会生成验证码，并且服务器会缓存本次验证码信息，并设置一个过期时间，最后将验证码返回给客户端，也就是向客户邮箱发送验证码邮件。

在用户填写并发送验证码给服务端之后，服务端会进行验证码校验，将客户发送的验证码与本地服务器缓存中的验证码信息进行对比，最后将结果返回给客户端。模块数据流图如下图3-4所示。

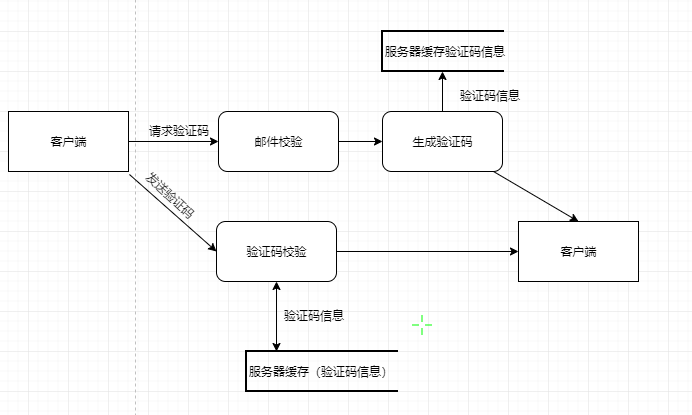


图 3‑4

### 3.1.4 文件上传模块

客户端将分片文件向服务端发送时，需携带token，经过身份认定之后，服务端接受分片文件，使用工具类对分片文件进行解析，并将分片文件临时存储在服务器磁盘中的临时分区，同时对每个分片文件进行基本信息校验，如当前分片数，文件名称，文件大小等。当最后一个分片文件到达时，服务端将开始对所有分片文件进行整合，使用缓冲输出流将每个小分片文件写入目标磁盘中，组合还原成原本的文件，同时组合好的视频文件会进行截取封面的流程，将视频的封面信息提取出来，保存到视频数据库中，并将文件以及图片信息返回给客户端。模块数据流图如下图3-5所示

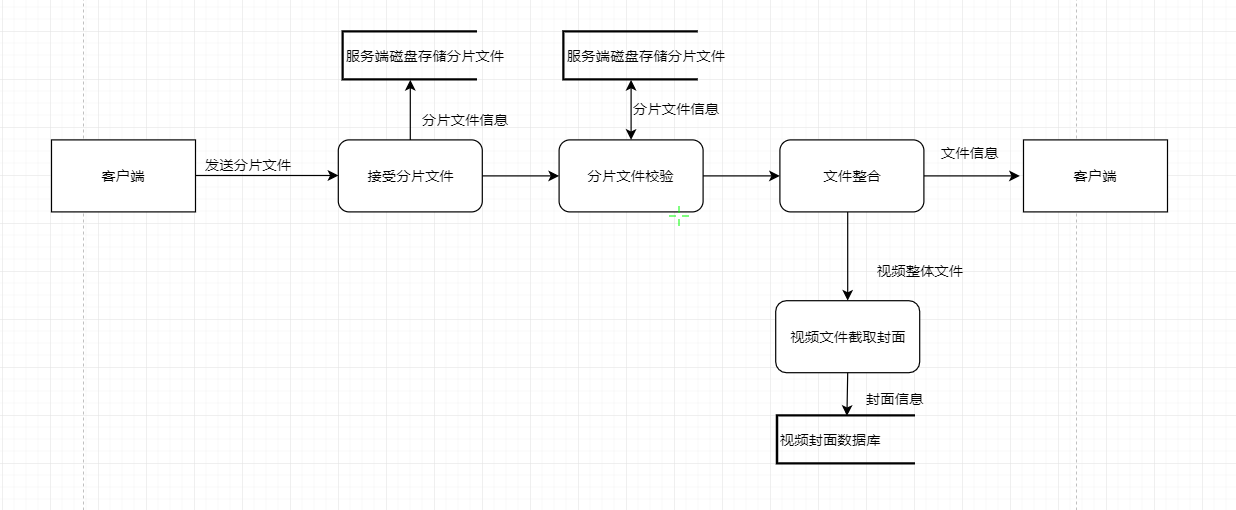


图 3‑5

### 3.1.5 习题组模块

习题组模块中包含选择题、判断题和主观题三个模块，客户端向服务端发送事务请求，如新增或是查询习题组。服务端接受事务后，如果是新增习题组请求，则会将即将要新增的习题组中的选择题、判断题和主观题id集合分别的对数据库进行信息查询，校验。校验完成后将本习题组信息存到数据库中，并且产生审核报表，将审核报表存入数据库中，并且将报表发给管理员端。若是查询请求，则按习题组id去数据库查询，若是已经审核通过的习题组信息才可发向客户端。模块数据流图如下图3-6所示。

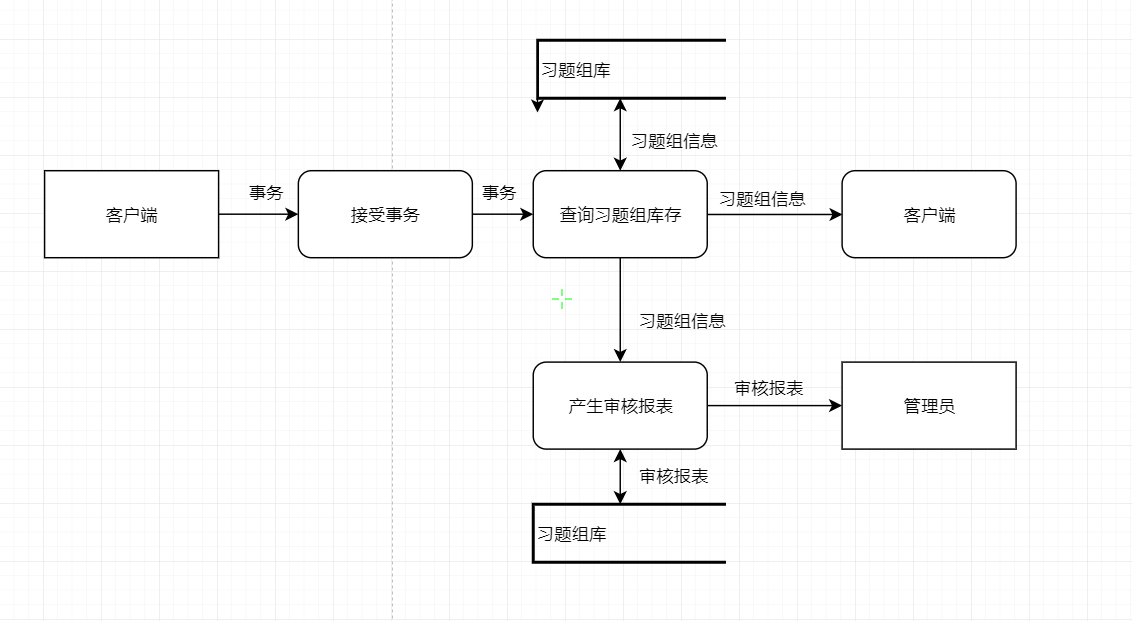


图 3‑6

### 3.1.6 选择题模块

客户端发送新增或更新选择题请求，服务端接受选择题信息之后，对用户信息与选择题信息进行整合，通过后将更新选择题库存清单，并在最后将选择题信息返回给客户端。模块数据流图如下图3-7所示。

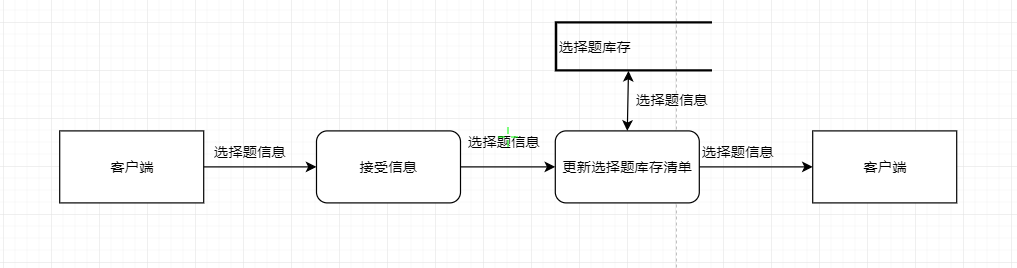


图 3‑7

### 3.1.7 判断题模块

客户端发送新增或更新判断题请求，服务端将判断题信息接收，在校验用户身份之后，与用户身份信息整合存入判断题库存，最后将判断题信息返回给客户端。模块数据流图如下图3-8所示。

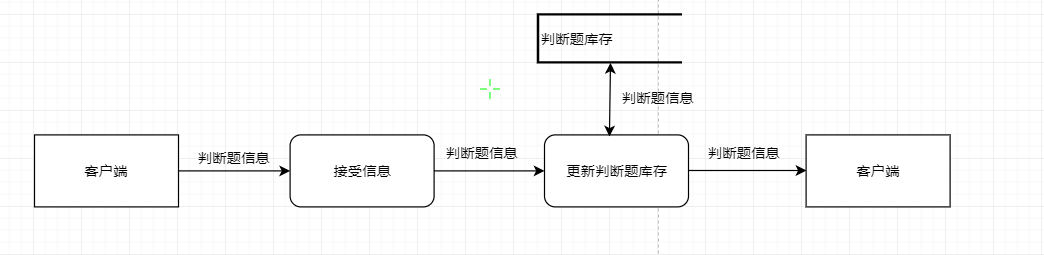


图 3‑8

### 3.1.8 主观题模块

客户端发送新增或更新主观题请求，服务端对主观题信息接收并做部分检查，如检查内容长度等，随后将用户信息与主观题信息整合并存入库存，最后将主观题信息返回客户端。模块数据流图如下图3-9所示。

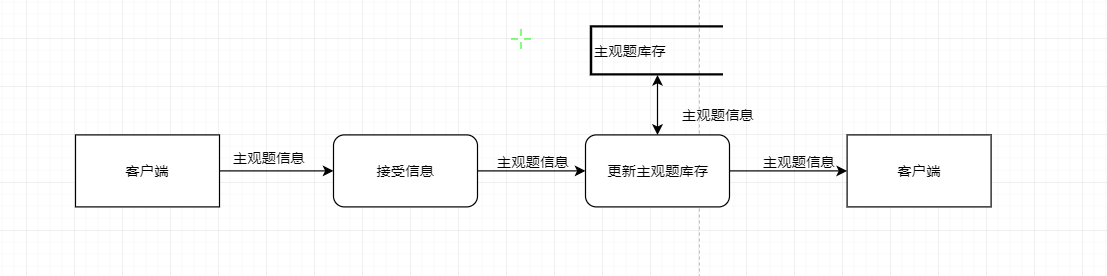


图 3‑9

### 3.1.9 视频模块

视频模块中的事务主要分为新增和查询两个主要，更新为辅，客户端发送新增视频的请求时，需要将视频文件上传完成，在发送新增请求时，服务端会解析客户端发送的视频信息，进行基本校验并且存入视频库存清单中，同时，也会产生一份审核报表给到管理员端，待管理员审核。若是客户端发送查询请求，则会根据发送的id信息进入视频校验中心校验，通过后进入视频库存中查询，最后将视频信息返回给客户端。模块数据流图如下图3-10所示。

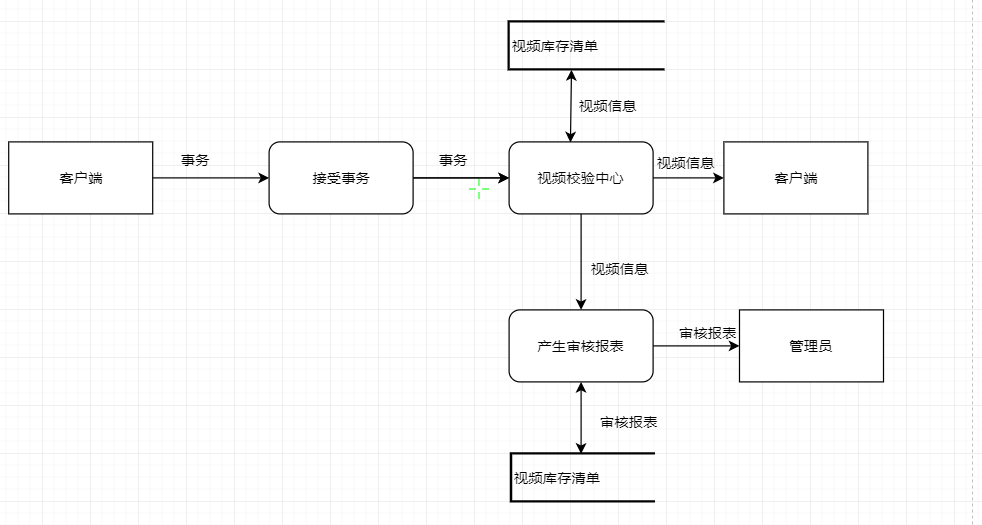


图 3‑0‑10

### 3.1.10 视频历史记录模块

在用户登录并且正在观看视频之时，客户端将用户token和视频当前观看时间每隔一定时间间隔向服务端发送，服务端接受观看信息，对用户发来的信息进行校验，确保用户为正常登录状态后，将客户端发来的观看时间进行基本校验，比如检验时间是否超出视频总时长等等校验，校验完成后，将信息存入MongoDB中，方便以后的大量读取和更新。最后将保存成功的消息发回客户端。模块数据流图如下图3-11所示。

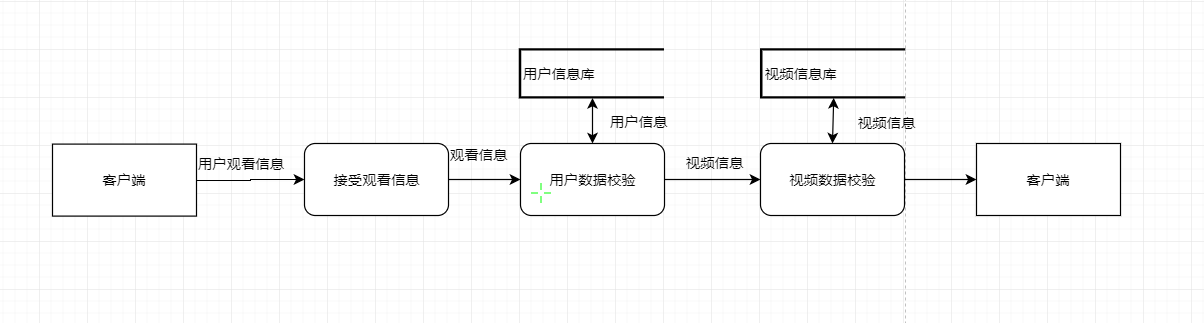


图 3‑0‑21

### 3.1.11 个人学习进度模块

当用户登录之后，看过的视频和做过提交的习题会记录在个人学习进度中，在个人学习进度模块中，每当有新增或是查询的请求时，服务端先对用户身份信息进行校验，校验完成之后，如果是新的视频记录则需要视频信息校验，将信息与视频信息库中的信息作对比。如果是新的习题记录则需要与习题信息库进行校验，再核对完成之后，整合的视频习题信息存入个人学习进度中。若是查询的请求，则按用户信息直接到用户历史信息校验部分，并且去对应的习题信息库和视频信息库中查询，得出的视频和习题信息经过整合之后返回给客户端。模块数据流图如下图3-12所示。

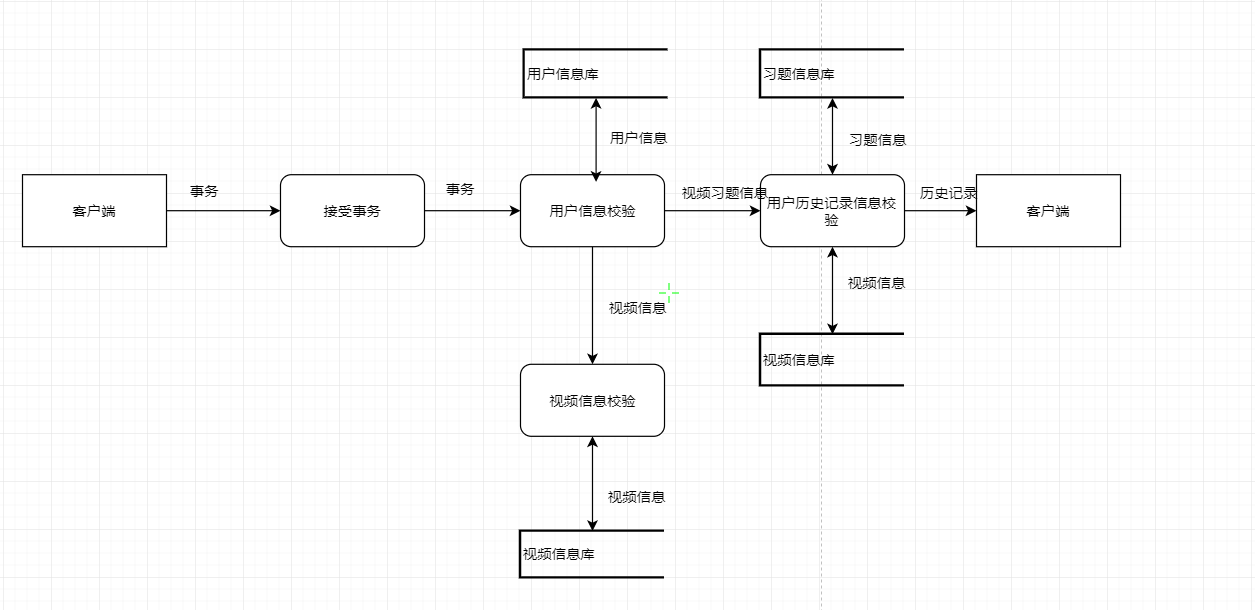


图 3‑0‑32

# 4 数据库设计

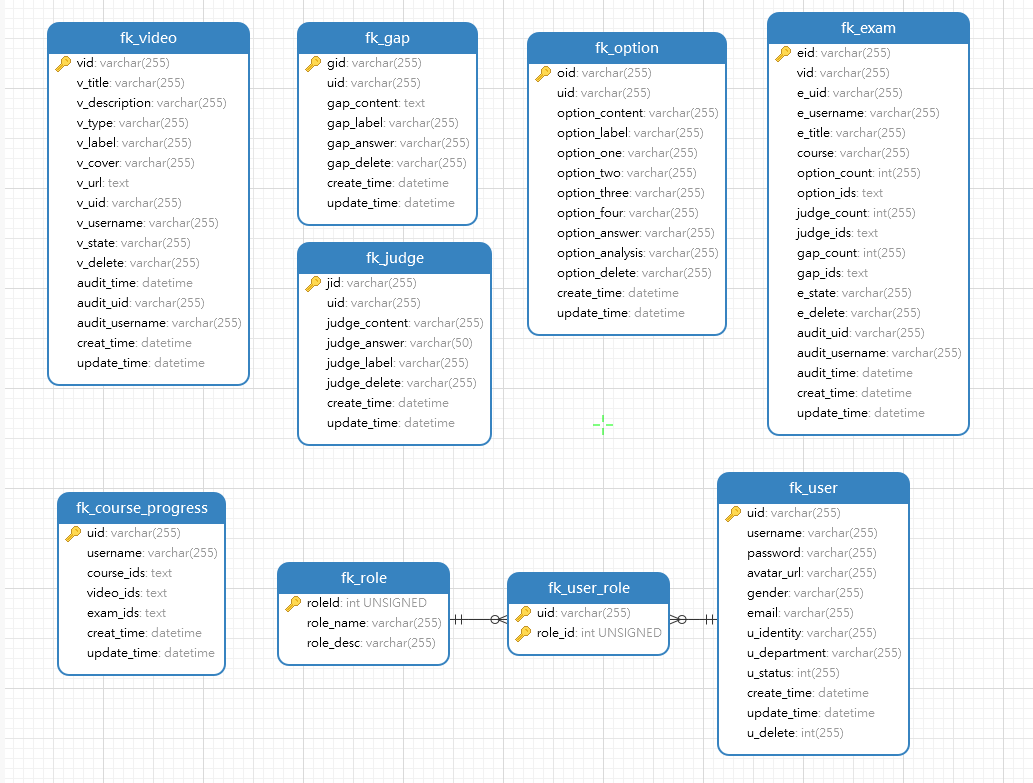
根据系统结构设计的功能分析，后端数据库部分主要的设计有，用户表、角色表、用户角色表、视频表、视频历史记录时间表、视频播放量表、习题组表、选择题表、判断题表、主观题表、个人学习进度表。使用的数据库有关系型MySQL和MongoDB非关系型数据库。

1. MySQL的优势

MySQL是常用的关系型数据库，运行速度快，MySQL体积小，执行命令快。使用成本低，MySQL是开源的，且提供免费版本，使用方便。历史悠久，使用稳定，性能卓越，占用资源少等等优点深受广大开发者喜爱[3]。

1. MongoDB的优势

MongoDB是一个开源、高性能、无模式的文档型数据库，是非关系型数据库产品中的一种。传统型关系型数据库在“三高”需求时力不从心，而MongoDB就能很好的应对数据库的高并发读写需求，对海量数据的高效率访问和存储的需求，还有就是对数据库的高可用性和高扩展性的需求。同时它所支持的数据结构非常松散，以其独有的一种类似于JSON的格式叫BSON，相当的灵活[2]。



## 4.1 用户表

本表主要存放一些用户的基本信息。用户uid为用户表主键，不允许为null，生成方式使用的是UUID随机生成，剩余的字段有username（用户名），password（密码），avatar\_url（头像），gender（性别），email（邮箱），u\_identity（身份），u\_department（学院），u\_status（状态），creat\_time（创建时间），update\_time（更新时间），u\_delete（是否删除）。其中，密码字段在保存用户密码时，不会以明文存储的方式，而是经过一定的加盐加密的方式存入，避免直接暴露密码。密码加密方式使用Spring Security框架提供的加密方式，大幅提高安全程度[4]。表格如下表格4-1所示。

表格 4‑0‑1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 是否主键 | 注释 |
| uid | varchar | 255 | 非null | 是 | 用户id |
| username | varchar | 255 | 非null |  | 用户名 |
| password | varchar | 255 | 非null |  | 密码 |
| avatar\_url | varchar | 255 |  |  | 用户头像 |
| gender | varchar | 255 | 非null |  | 性别 |
| email | varchar | 255 | 非null |  | 邮箱 |
| u\_identity | varchar | 255 | 非null |  | 身份 |
| u\_department | varchar | 255 | 非null |  | 学院 |
| u\_status | int | 255 | 非null |  | 状态（1正常，0封禁） |
| creat\_time | datetime |  | 非null |  | 创建时间 |
| update\_time | datetime |  | 非null |  | 更新时间 |
| u\_delete | int | 255 | 非null |  | 是否删除（1正常，0删除） |

## 4.2 角色表

本表为角色信息表，主要保存角色的名字，身份，描述等。字段有roleId（角色id），role\_name（角色名字），role\_desc（描述）三个字段。是为了区分普通用户以及管理员的角色所使用的表，以及为了控制不同用户的操作权限，此表还可额外扩展，可以加入不同等级的角色来扩展系统不同用户的操作权限。表格如下表格4-2所示。

表格 4‑0‑2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 是否主键 | 注释 |
| roleId | int | 255 | 非null | 主键 | 角色id |
| role\_name | varchar | 255 | 非null |  | 角色名字，身份 |
| role\_desc | varchar | 255 | 非null |  | 描述 |

## 4.3 用户角色表

此表为中间链接表，此表用户uid为用户表uid的外键，role\_id为角色表roleId外键。本表作用是记录所有已经注册的用户所含有的角色信息，以及所拥有的权限等级。表格如下表格4-3所示。

表格 4‑0‑3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 是否主键 | 注释 |
| uid | varchar | 255 | 非null | 主键 | 用户id |
| role\_id | int | 255 | 非null | 主键 | 角色id |

## 4.4 视频表

本表是存放用户所上传的所有视频内容的基本信息，有vid（视频id），v\_title（视频标题），v\_description（视频简介），v\_type（视频分类）,v\_label（视频标签），v\_cover（视频封面），v\_url（存放上传视频的地址），v\_uid（上传人id），v\_username（上传人用户名），v\_state（审核状态 0,：上传成功（待提交审核）,1：审核不通过，2：审核通过，3：发布，4：下架，），v\_delete（是否删除 1：正常 0：删除），audit\_time（视频审核时间），audit\_username（审核人用户名），creat\_time（上传时间），update\_time（更新时间）。表格如下表格4-4所示。

表格 4‑0‑4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 是否主键 | 注释 |
| vid | varchar | 255 | 非null | 主键 | 视频id |
| v\_title | varchar | 255 | 非null |  | 视频标题 |
| v\_description | varchar | 255 | 非null |  | 视频描述（简介） |
| v\_type | varchar | 255 | 非null |  | 视频分类 |
| v\_label | varchar | 255 |  |  | 视频标签 |
| v\_cover | varchar | 255 | 非null |  | 视频封面 |
| v\_url | text |  | 非null |  | 上传的视频地址 |
| v\_uid | varchar | 255 | 非null |  | 上传人id |
| v\_username | varchar | 255 | 非null |  | 上传人用户名 |
| v\_state | varchar | 255 | 非null |  | 审核状态 |
| v\_delete | varchar | 255 | 非null |  | 是否删除 |
| audit\_time | datetime |  |  |  | 视频审核时间 |
| audit\_username | varchar | 255 |  |  | 审核人用户名 |
| creat\_time | datetime |  | 非null |  | 上传时间 |
| update\_time | datetime |  | 非null |  | 更新时间 |

## 4.5 视频历史记录时间表

此表主要保存用户在进行视频观看后所保存的一些上次观看时间信息。字段有videoTimeId（历史记录id），vid（视频id），uid（观看用户id）和 playTime（视频上次观看进度）。由于playTime这个字段是一个需要频繁更新的字段，所以本视频记录时间表是放在MongoDB中使用。表格如下表格4-5所示。

表格 4‑0‑5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 是否主键 | 注释 |
| videoTimeId | varchar | 255 | 非null | 主键 | 历史记录id |
| uid | varchar | 255 | 非null |  | 观看用户id |
| vid | varchar | 255 | 非null |  | 视频id |
| playTime | Varchar | 255 | 非null |  | 视频上次进度 |

## 4.6 视频播放量表

此表为视频播放量表，用于记录视频的播放量，其中vid为视频id作为主键，vUid为上传本视频的用户id，使用此字段方便于后续的查找信息。videoCount为视频播放量，用于记录播放次数。表格如下表格4-6所示。

表格 4‑0‑6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 是否主键 | 注释 |
| vid | varchar | 255 | 非null | 是 | 视频id |
| vUid | varchar | 255 | 非null |  | 上传用户id |
| videoCount | int | 255 | 非null |  | 观看数 |

## 4.7 习题组表

本表为习题组表，主要作用是保存一整个习题中所有的题目数量和id集合，其中审核状态e\_state（待审核，通过），是否删除字段e\_delete（1正常，0删除）。之所以有些字段选用text是因为考虑到习题的id集合长度可能会较长，容易超出长度限制。表格如下表格4-7所示。

表格 4‑0‑7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 是否主键 | 注释 |
| eid | varchar | 255 | 非null | 主键 | 试卷id |
| vid | varchar | 255 |  |  | 跟随的视频id |
| e\_uid | varchar | 255 | 非null |  | 上传人id |
| e\_username | varchar | 255 | 非null |  | 上传人用户名 |
| e\_title | varchar | 255 | 非null |  | 习题组标签 |
| course | varchar | 255 |  |  | 课程名字 |
| option\_count | int | 255 | 非null |  | 选择题数 |
| option\_ids | text |  |  |  | 所有选择题id |
| juedge\_count | int | 255 | 非null |  | 判断题数 |
| judge\_ids | text |  |  |  | 所有判断题id |
| gap\_count | int | 255 | 非null |  | 主观题数 |
| gap\_ids | text |  |  |  | 所有主观题id |
| e\_state | varchar | 255 | 非null |  | 审核状态 |
| e\_delete | varchar | 255 | 非null |  | 是否删除 |
| audit\_uid | varchar | 255 |  |  | 审核人id |
| audit\_username | varchar | 255 |  |  | 审核人用户名 |
| audit\_time | datetime |  |  |  | 审核时间 |
| creat\_time | datetime |  | 非null |  | 创建时间 |
| update\_time | datetime |  | 非null |  | 更新时间 |

## 4.8 选择题表

本表为选择题表，存放选择题内容，其中是否删除字段option\_delete（1正常，0删除），答案解析可以为略。表格如下表格4-8所示。

表格 4‑8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 是否主键 | 注释 |
| oid | varchar | 255 | 非null | 主键 | 选择题id |
| uid | varchar | 255 | 非null |  | 上传人id |
| option\_content | varchar | 255 | 非null |  | 选择题内容 |
| option\_label | varchar | 255 |  |  | 选择题标签 |
| option\_one | varchar | 255 | 非null |  | 选项一 |
| option\_two | varchar | 255 | 非null |  | 选项二 |
| option\_three | varchar | 255 | 非null |  | 选项三 |
| option\_four | varchar | 255 | 非null |  | 选项四 |
| option\_answer | varchar | 255 | 非null |  | 答案 |
| option\_analysis | varchar | 255 |  |  | 答案解析 |
| option\_delete | varchar | 255 | 非null |  | 是否删除 |
| creat\_time | datetime |  | 非null |  | 创建时间 |
| update\_time | datetime |  | 非null |  | 更新时间 |

## 4.9 判断题表

本表为判断题表，主要存储判断题信息。表格如下表格4-9所示。

表格 4‑9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 是否主键 | 注释 |
| jid | varchar | 255 | 非null | 主键 | 判断题id |
| uid | varchar | 255 | 非null |  | 上传人id |
| judge\_content | varchar | 255 | 非null |  | 题目内容 |
| judge\_answer | varchar | 50 | 非null |  | 答案 |
| judge\_label | varchar | 255 |  |  | 标签 |
| judge\_delete | varchar | 255 | 非null |  | 是否删除 |
| creat\_time | datetime |  | 非null |  | 创建时间 |
| update\_time | datetime |  | 非null |  | 更新时间 |

## 4.10 主观题表

本表为主观题表，主要存放主观题信息。考虑到主观题的题目内容，所以使用text类型来存储，可以放下较长的题目信息。表格如下表格4-10所示。

表格 4‑10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 是否主键 | 注释 |
| gid | varchar | 255 | 非null | 主键 | 主观题id |
| uid | varchar | 255 | 非null |  | 上传人id |
| gap\_content | text |  | 非null |  | 题目内容 |
| gap\_label | varchar | 255 |  |  | 主观题标签 |
| gap\_answer | varchar | 255 | 非null |  | 答案 |
| gap\_delete | varchar | 255 | 非null |  | 是否删除 |
| creat\_time | datetime |  | 非null |  | 创建时间 |
| update\_time | datetime |  | 非null |  | 更新时间 |

## 4.11 个人学习进度表

本表为个人学习进度表，主要用于展示个人学习进度，将看过的视频和做过的习题组都放在本表。考虑到id集合可能较长，所以使用text类型存储。表格如下表格4-11所示。

表格 4‑11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 是否主键 | 注释 |
| uid | varchar | 255 | 非null | 主键 | 上传人id |
| username | varchar | 255 | 非null |  | 用户名 |
| course\_ids | text |  |  |  | 学过的课程集合 |
| video\_ids | text |  |  |  | 学过的视频集合 |
| exam\_ids | text |  |  |  | 做过的习题组集合 |
| creat\_time | datetime |  | 非null |  | 创建时间 |
| update\_time | datetime |  | 非null |  | 更新时间 |

# 5 系统实现

## 5.1 登录实现

本系统采用邮箱作为账号登录，具体如下图5-1所示，在输入完邮箱和密码点击登录之后，前端将登录表单数组化，然后转化成JSON格式，封装完成后发送Ajax请求，请求发到后端被JwtLoginFilter拦截，使用Spring Security框架进行登录验证，若密码错误，则返回登录失败，通过认证之后，则使用Jwt工具包生成一个token，同时查询一次数据库，将登录用户的基本信息查出，封装到一个map集合中。将token和userMap以键值对的形式封装到Redis中保存用户信息，并且设置一个三十分钟的登录时间，防止占用缓存过久而导致服务器压力过大。最后服务器将token和用户基本信息返回给前端，前端在接受到登录成功的信息时，前端使用localStorage.setItem方法将token和用户基本信息保存在网页中，以方便后续的请求访问。在登录成功之后，前端会提取用户信息，提取用户名并在页面最上方显示：欢迎回来+用户名，同时判断用户是否为管理员，若为管理员，则在页面最上方将会增添一个管理员界面的点击入口。



图 5‑0‑1

## 5.2 注册实现

注册功能如下图5-2所示，其中，密码填写需要8-20位英文或数字，邮箱和密码在输入完后，前端会有一个校验，判断邮箱格式是否正确和密码是否按规范填写。当点击发送验证码按钮时，按钮进入倒计时，倒计时未结束前无法再次发送，同时前端将未注册的用户邮箱发往后端，后端中的Spring Security经过配置，不拦截本次请求，后端接口接收到邮件之后。生成一个验证码，并且将验证码和邮箱以键值对的形式放入Redis缓存之中，并且设置一定的过期时间，将验证码封装到一封邮件之中，以邮件的形式发送到用户的邮箱之中。

用户将从邮箱中获取的验证码，输入到前端，并且点击注册按钮时，前端将表单数据封装成JSON格式，后端接收后首先校验验证码，然后校验昵称和邮箱是否被注册，最后注册成功。此时用户密码将会被Spring Security的passwordEncoder.encode方法加密保存到数据库。

使用Redis的优势有，它是一款开源并且高性能的缓存数据库，通过键值对的形式存储。其中的Value值还支持多种数据结构的存储。数据存在内存中，直接与内存连接，并且支持键过期功能，实现定时缓存，本系统就是使用了这个优点，可以及时清楚过期Key值，减少Redis的内容空间占用[5]。



图 5‑2

## 5.3 视频列表

视频列表功能如下图5-3-1和图5-3-2所示。其中当点击视频总览时，跳转到视频展示页面，每一页中含有20个视频整齐排列，其中主要内容为计算机组成原理的一些学习视频。页面最上方为查询功能，可以按视频标题、视频简介和上传视频用户名查询。后端查询出视频信息之后封装到一个DTO对象中，DTO为数据传输对象，好处就是继承了一些实体类的内容，但又不至于将一些敏感信息直接暴露在前端，提升了整体的安全性。在完成封装DTO之后，将会封装一个map集合，其中包含当前页码，总页码，总条数，以及所有的视频等。最后在前端进行结果展示。



图 5‑3-1

其中，每一个视频都是一个可以点击的链接，点击之后进入到具体的视频播放页面，在每一个视频的图标里，包含了视频封面、视频作者、视频标题和视频上传时间等信息，方便用户更好初步了解视频内容。

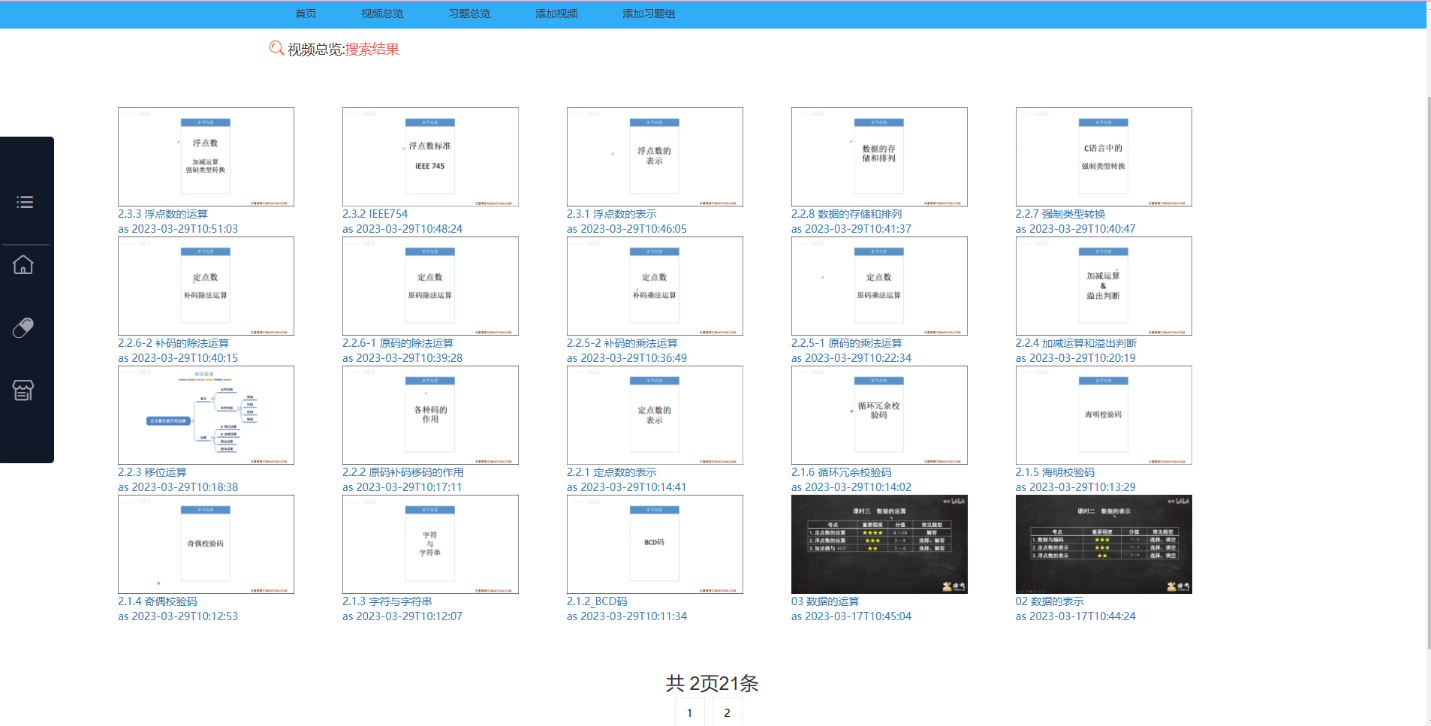


图 5‑3-2

## 5.4 视频播放页面

页面如图5-4所示。当进入到视频播放页面时，前端首先从URL中提取出视频编号id，后端在接收到请求之后，根据前端发来的视频id进行查找，将视频信息经过封装之后发回前端。与之同时前端发送增加视频播放量的请求到后端，后端则提升对应的视频播放量。前端收到视频信息之后，使用ckplayer工具初始化播放器，并且将视频作者、视频分区、视频播放量、视频标题、视频简介显示到前端。如果用户处于登录状态，则会在用户播放视频的过程中，每过几秒，前端将会通过ckplayer工具获取到用户当前视频进度，前端携带用户token并且将此进度发向后端，后端会在VerifyFilter中拦截并且获取用户token，随后会验证token是否正确，此时Redis中的用户信息的过期时间会被刷新，如果是错误的token则会给用户提示。随后将前端的视频进度添加或者更新到MongoDB中的视频进度表。如果视频观看结束，前端将会往后端中个人学习进度的模块发送请求，后端在校验token之后，将认定此视频已被用户学习完毕，将会添加到个人学习进度的记录中。

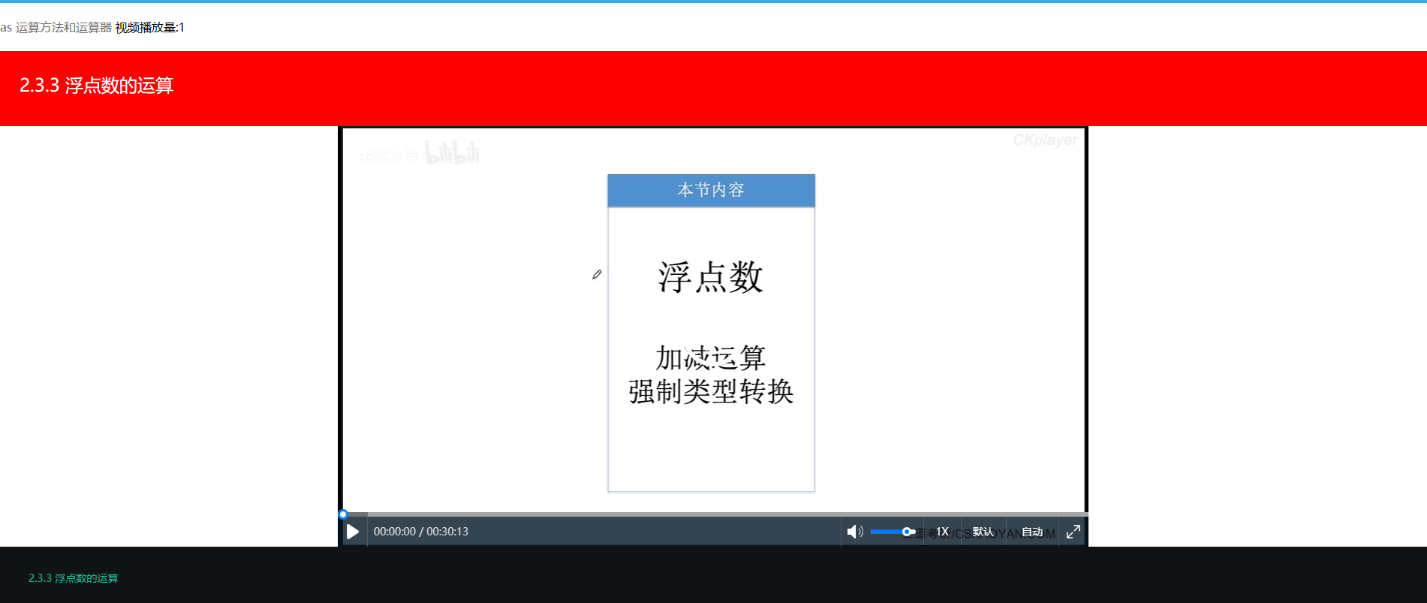


图 5‑4

## 5.5 习题总览界面

界面如图5-5-1和图5-5-2所示。前端发送查询请求之后，首先会查看搜索框中是否有查询习题的搜索。后端接到请求之后，将会把已经审核通过的习题按时间倒序的方式排列发到前端展示，如果有搜索关键词，则会根据习题组标题、课程来查询，将习题信息进行封装，最后发到前端。

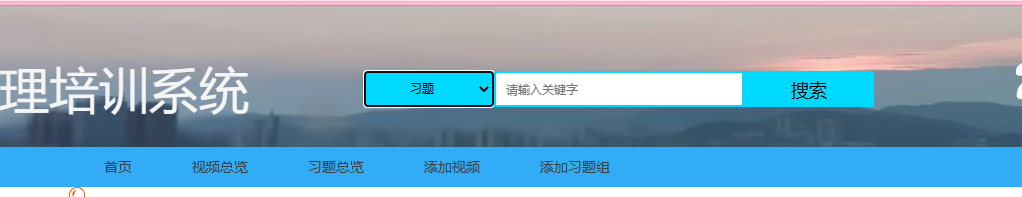


图 5‑5-1

其中，可以很清楚的看到，选择题数量，判断题数量和主观题数量。



图 5‑5-2

## 5.6 习题界面

如图5-6所示。在习题页面中，将会具体的展示所有题目的内容，其中后端会根据前端发来的习题id去查找具体的内容，根据习题组中的题目信息去查找，将其中的选择题、判断题和主观题全部查询出来并且统一封装发到前端。其中，习题界面可以直接查看答案，答案按钮放在每道题对应的地方，里面有出题人撰写的答案和答案解析。若用户点击提交按钮，则会携带token向后端请求，校验成功之后，将会在用户的个人学习进度中保留已完成本习题记录。



图 5‑6

## 5.7 视频上传界面

界面如下图5-7-1和5-7-2所示。视频在选择文件之后就同步上传。视频文件上传使用了Webuploader文件分片上传技术，还可以做到断点续传的功能。文件，特别是大文件在上传的过程中总会遇到上传失败的问题，原因可能是网速，网络延迟，也可能是读取速度太慢到时请求超时，而且通常遇到上传失败时总是会重复上传，可能会造成资源重复，而分片上传和断点续传就可以很好的解决这个问题，大大的提高了文件传输速度和安全性，提高了系统并发能力[6]。



图 5-7-1

在选择视频文件之后，前端还是同步上传视频文件，通过WebUploader工具将视频文件分成多个小文件，多个线程同时发送到后端，并且对视频格式进行了一定限制，在视频上传之前，还会先检验用户的token是否过期，以确保用户处于登录状态。后端在接收到一个个分片小文件时，先开辟一个临时目录进行初步存储，并且获取当前分片数和总分片数，在临时目录中对每个分片小文件进行标号。在最后一个分片文件发送时，后端生成一个缓冲输出流，将需要保存的文件目标目录放入缓冲输出流中，并开始遍历每一个分片，如果文件不存在，则等待文件存在为止，将每个分片小文件转化成byte数组，此时缓冲输出流将byte数据写入到最终目标文件中。最后文件组合成功时，调用截取视频封面方法，通过FFmpegFrameGrabber的方法截取视频的某一帧为作为视频封面。所有信息整合完毕将信息发回前端。前端则会将视频封面放到对应位置，剩余信息由用户填写好后，提交即可保存成功如图中所示。后端将会把视频的基本信息和用户填写的信息保存到视频数据库中。



图 5-7-2

## 5.8 添加习题

如图5-8-1和5-8-2所示。首先是未添加习题的初始界面。

图 5-8-1

当每个添加按钮被点击时，对应的题目开始计数，并且记录用户所填内容，例如选择题，其中当填完内容时，需要及时点击保存按钮以防丢失，其他题目类似。多次点击在后端会持续的更新每一道题所填内容。在填完所有题目点击发布时，前端会整合所有题目信息，将信息打包发送到后端进行解析，后端会判断题目是否存在，随后将习题组保存到数据库中。

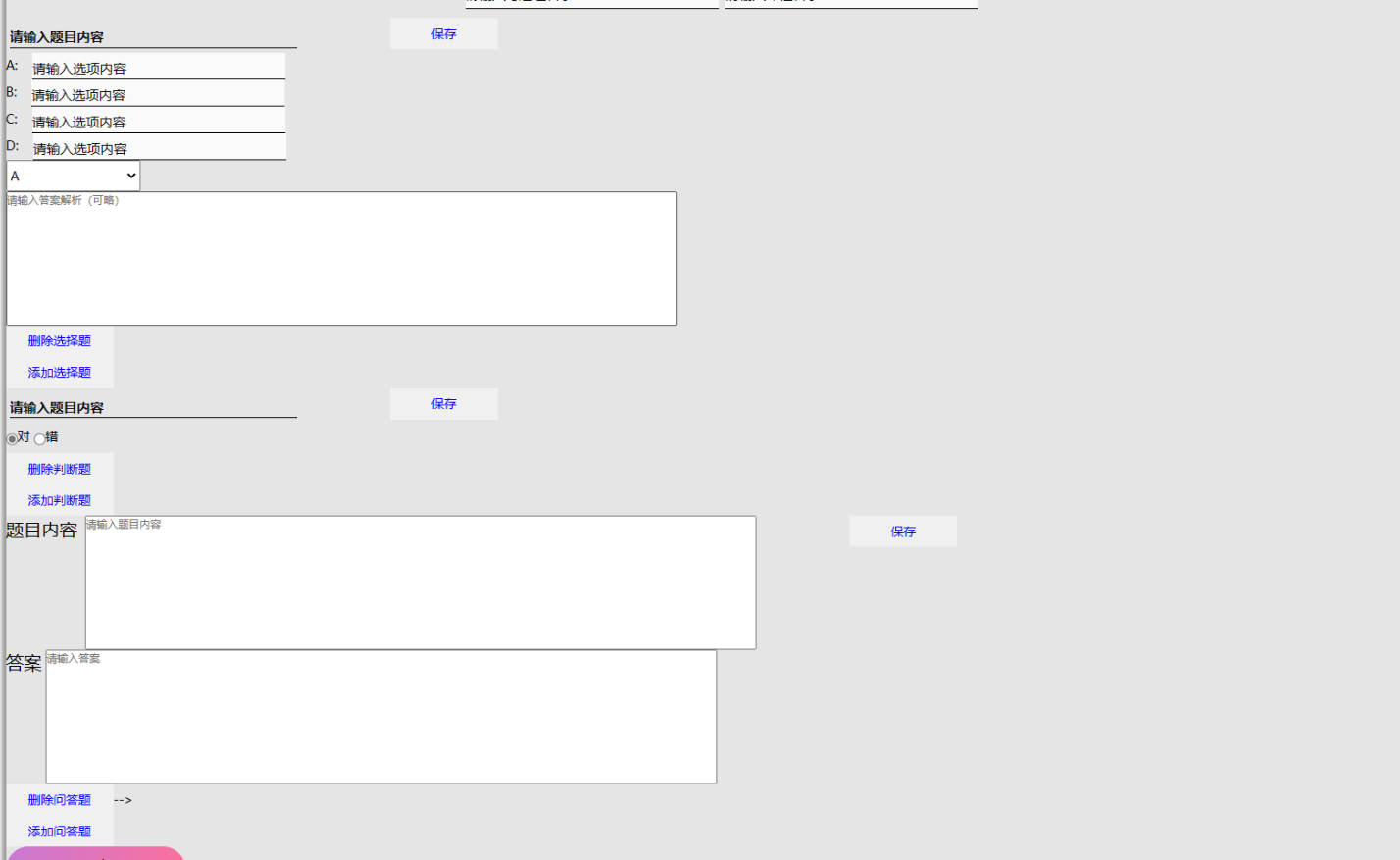


图 5-8-2

## 5.9 首页

首页如下图5-9所示。首页主要展示一些基本的计算机组成原理课程的轮播图，课程介绍，课程目标等，其中还有一些推荐视频放到首页内展示，侧边栏有快速访问的按钮。



图 5-9

## 5.10 个人主页

如图5-10-1和5-10-2和5-10-3所示，第一部分主要展示用户的基本信息，如邮箱、性别、身份、用户名等内容。



图 5-10-1

本部分展示该用户所学过的视频，有看过的记录都会放在个人主页之中，通过本功能，可以更好的了解到学到哪一步，并且对自己的下一步学习有一个更好的，更清晰的判断。习题部分也是同理，已完成的习题将会放在个人主页的记录中。



图 5-10-2

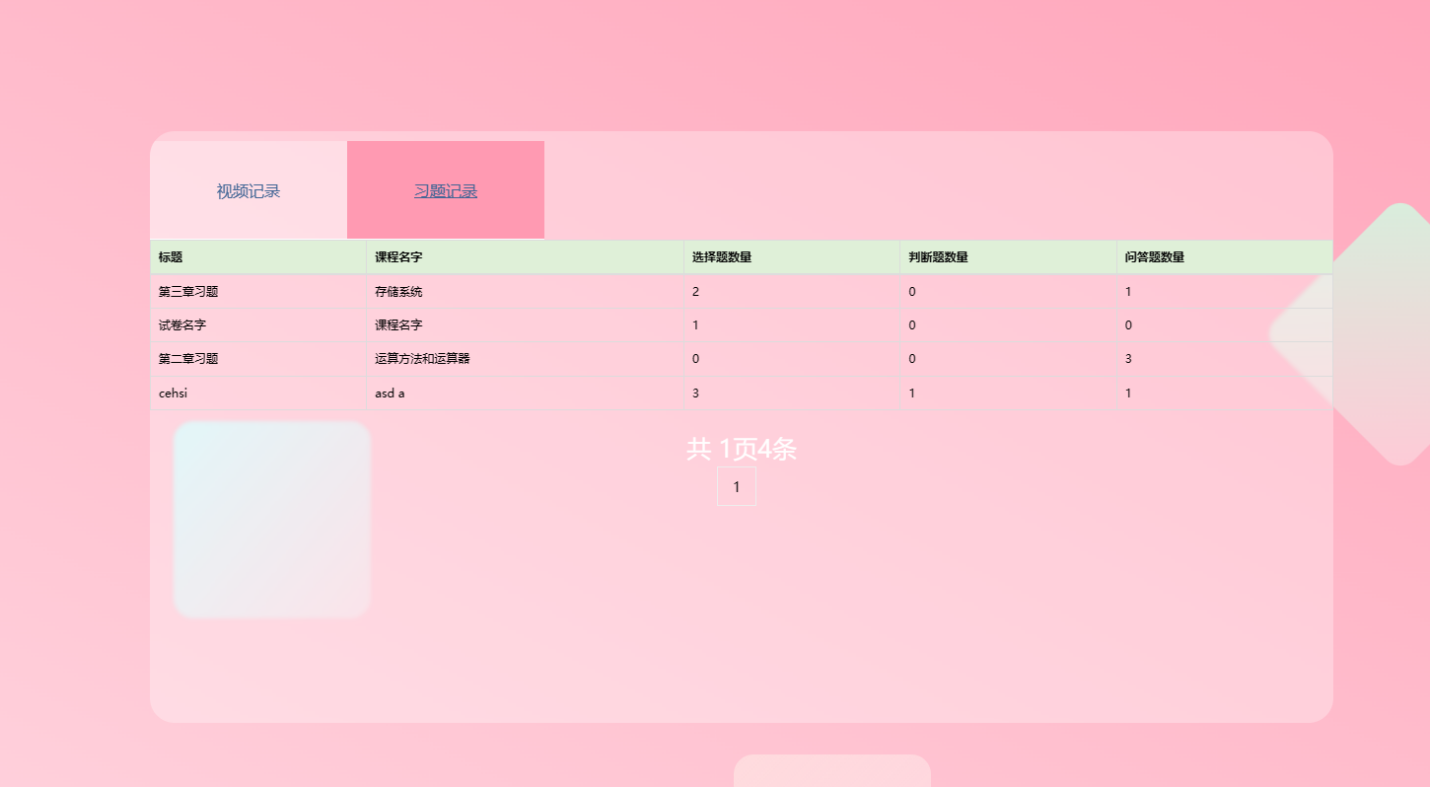


图 5-10-3

## 5.11 管理员界面

首先如图5-11-1所示。一进入到管理员界面首先是视频审核界面，这里可以看到所有刚上传的视频，其中若是要进行具体审核，可以点击视频编号，页面将会跳转到视频的具体播放页面，管理员可以对视频内容进行查看。此页面中，将视频审核的一些基本信息都有所展示。本界面同样拥有查询和分页功能。



图 5-11-1

如图5-11-2所示，当审核完成之后，管理员可以点击修改按钮，此时会跳出来一个弹窗，管理员可以更改视频的审核状态，若是想让视频能正常的在首页展示，则需要审核通过才可以。

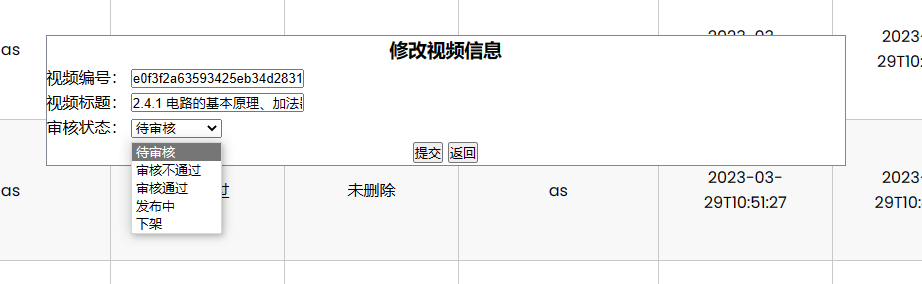


图 5-11-2

如图5-11-3所示，此为习题管理界面，习题审核大致与视频审核一致，其中点击习题组编号同样可以进入到具体的页面查看，审核。

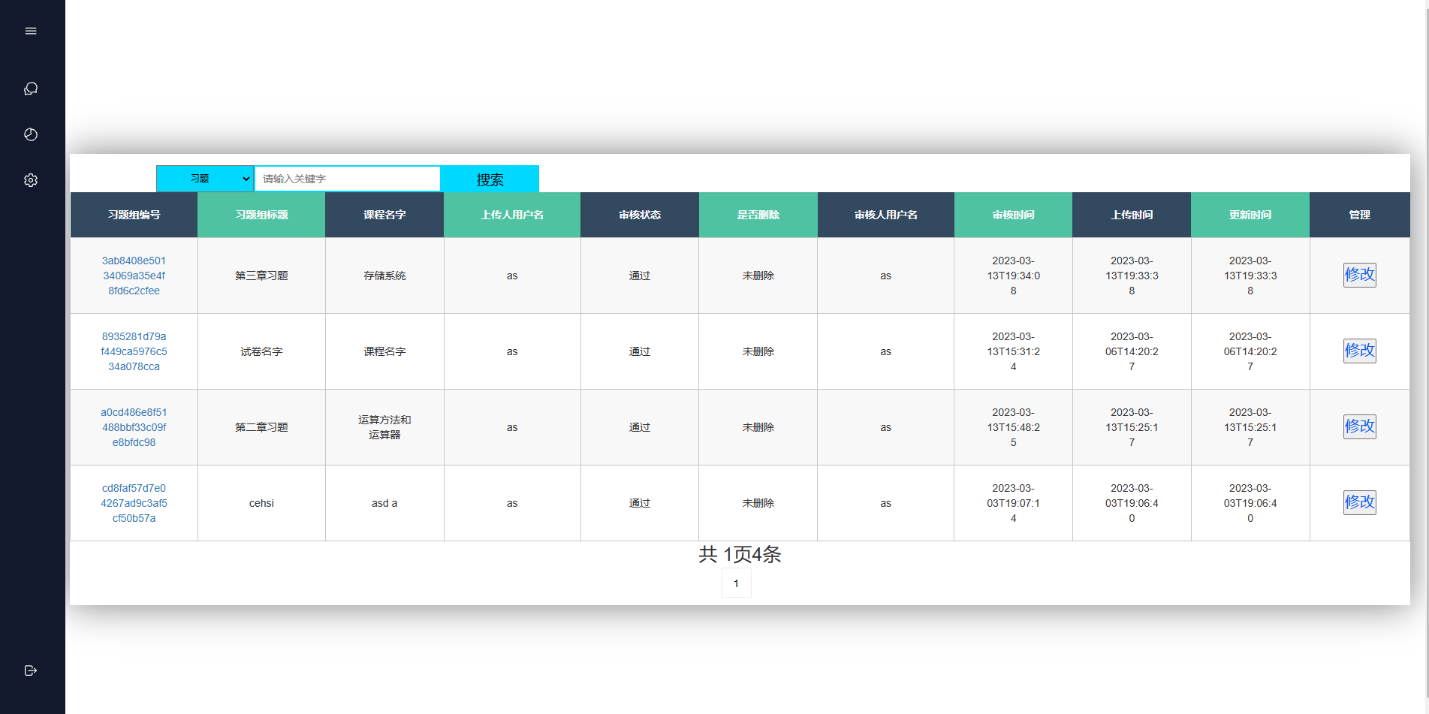


图 5-11-3

如图5-11-4所示，当审核完成时，点击修改按钮即可跳出本窗口，在窗口中修改审核状态即可。

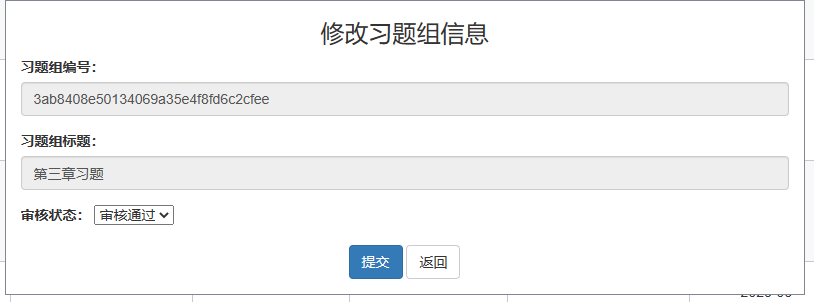


图 5-11-4

# 6 结论

本系统实现了发布视频，发布习题，管理员审核，视频播放以及观看，做习题，视频进度的保存，学习进度保存等功能，尤其是视频进度保存和个人学习进度的保存，可以让学生们明确的知道自己学到具体哪一步，并针对自己的学习展开下一步计划。本系统达到了看视频学习，并且做习题练习的目的，可以达到拓宽学生们学习计算机组成原理课程的来源，多了一种学习方式。并且达到在线上使用本系统进行课前预习和课后复习，对较难的知识点有充足的时间可以学习，也能提升学生学习本课程的自主性和积极性。

参考文献:

1. 伍超平. 基于教案-习题-视频三循环交互的在线学习平台设计[D].天津师范大学,2020.DOI:10.27363/d.cnki.gtsfu.2020.000150.辛希孟, 陈绍业, 王凤翥. 信息技术与信息服务研讨[A]. 杨洋. 中华国际论文集 [C]. 北京: 中国社会科学出版社, 1994: 12-18.
2. 贾成果.MongoDB三种集群方案研究及性能对比分析[J].中国传媒科技,2020,No.329(08):115-116+128.DOI:10.19483/j.cnki.11-4653/n.2020.08.030.
3. 张丽景,张文川.基于对比分析法的高职“MySQL数据库”课程设计——以兰州石化职业技术大学为例[J].现代信息科技,2022,6(21):188-191.DOI:10.19850/j.cnki.2096-4706.2022.21.046.
4. 杜成龙,曹海平,吴亮.基于Spring Security的自定义用户认证和授权策略研究[J].信息系统工程,2022,No.345(09):75-78.
5. 张蒙蒙,曹成茂.大数据时代Redis缓存的性能分析与优化[J].巢湖学院学报,2022,24(03):80-87+96.
6. 张立伟,李涪帆.基于java语言的大文件分片传输[J].通讯世界,2020,27(06):51+53.
7. 林玲,许东耀.计算机硬件课程改革方法分析——以计算机组成原理课程为例[J].数字技术与应用,2023,41(02):85-87.DOI:10.19695/j.cnki.cn12-1369.2023.02.21.
8. 王海芸,张晓荔.知识类短视频对知识学习的影响[J].科教导刊,2022,No.498(30):52-54.DOI:10.16400/j.cnki.kjdk.2022.30.017.
9. 周公平.基于jQuery框架的Web前端开发设计方法研究[J].信息与电脑(理论版),2022,34(05):128-130.
10. 王海芸,张晓荔.知识类短视频对知识学习的影响[J].科教导刊,2022,No.498(30):52-54.DOI:10.16400/j.cnki.kjdk.2022.30.017.
11. Mahyurianti and Mahyuddin Nenny. The Application of Audio Visual Media Based on Minangkabau Folklore to Develop Children’s Speaking Ability in Kindergarten Kartika 1–7 Padang[C]. Padang, Indonesia, 2020.
12. Choe Ronny C et al. Student Satisfaction and Learning Outcomes in Asynchronous Online Lecture Videos.[J]. CBE life sciences education, 2019, 18(4) : ar55.

致 谢

每每遇到困难的时候，总是老师教导我不要放弃，也有同学们的有爱互助，还有自己的努力拼搏。感谢肇庆学院四年来对我的辛勤培育。

本论文是在我的指导老师的悉心指导下完成的。从课题的选择以及中期的检查到项目的最终完成，王俊波老师都始终给予我细心的指导和不懈的支持。在此谨向王俊波老师致以诚挚的谢意和崇高的敬意！

附录A

附A1

登录模块

@Override  
*public Authentication* attemptAuthentication ( *HttpServletRequest* request, *HttpServletResponse* response ) *throws* AuthenticationException  
{  
 *try* {  
 User user = *new* ObjectMapper().readValue(request.getInputStream(), User.*class*);  
 UsernamePasswordAuthenticationToken authRequest = *new* UsernamePasswordAuthenticationToken(user.getEmail(),user.getPassword());  
 *return this*.getAuthenticationManager().authenticate(authRequest);  
 }  
 *catch* ( Exception e )  
 {  
 *try* {  
 response.setContentType("application/json;charset=utf-8");  
 response.setStatus(*HttpServletResponse*.*SC\_UNAUTHORIZED*);  
 *Map*<Object, Object> resultMap = *new* HashMap<>(16);  
 resultMap.put("code",*HttpServletResponse*.*SC\_UNAUTHORIZED*);  
 resultMap.put("msg","用户名或密码错误");  
 ServletOutputStream out = response.getOutputStream();  
 OutputStreamWriter ow = *new* OutputStreamWriter(out, StandardCharsets.*UTF\_8*);  
 ow.write(*new* ObjectMapper().writeValueAsString(resultMap));  
 ow.flush();  
 ow.close();  
 }  
 *catch* ( Exception ex )  
 {  
 *throw new* RuntimeException(ex);  
 }  
 *throw new* RuntimeException(e);  
 }  
}

*/\*\*  
 \* 认证成功回写token  
 \*/*@Override  
*public void* successfulAuthentication ( *HttpServletRequest* request, *HttpServletResponse* response, *FilterChain* chain, *Authentication* authResult )  
{  
 User user = *new* User();  
 user.setUsername(authResult.getName());  
 user.setRoles((*List*<Role>) authResult.getAuthorities());  
 *//一天的时间* String token = JwtUtils.*generateTokenExpireInMinutes*(user, rsaKeyConfig.getPrivateKey());  
 *//将token写入redis缓存，需要查询一次数据库* LambdaQueryWrapper<User> lqw = *new* LambdaQueryWrapper<>();  
 lqw.eq(User::getUsername, user.getUsername());  
 User selectOne = userService.getOne(lqw);  
 UserDTO userDTO = BeanUtil.*copyProperties*(selectOne, UserDTO.*class*);  
 *List*<Role> roles = roleMapper.findUserRole(userDTO.getUid());  
 userDTO.setRoles(roles);  
 UserHolder.*saveUser*(userDTO);  
 *Map*<String, Object> userMap = BeanUtil.*beanToMap*(userDTO, *new* HashMap<>(16),  
 CopyOptions.*create*()  
 .setIgnoreNullValue(*true*)  
 .setFieldValueEditor(( fieldName, fieldValue ) -> String.*valueOf*(fieldValue)));  
 *//存入redis并设置token有效期,登录后还需要更新token* String tokenKey = RedisConstants.*LOGIN\_USER\_KEY* + token;  
 response.addHeader("authorization", "Bearer " + token);  
 stringRedisTemplate.opsForHash().putAll(tokenKey, userMap);  
 stringRedisTemplate.expire(tokenKey, RedisConstants.*LOGIN\_USER\_TTL*, TimeUnit.*MINUTES*);  
 *try* {  
 response.setContentType("application/json;charset=utf-8");  
 response.setStatus(*HttpServletResponse*.*SC\_OK*);  
 PrintWriter writer = response.getWriter();  
 *Map*<Object, Object> resultMap = *new* HashMap<>(16);  
 resultMap.put("code", *HttpServletResponse*.*SC\_OK*);  
 resultMap.put("user", userDTO);  
 resultMap.put("msg", "认证通过");  
 writer.write(*new* ObjectMapper().writeValueAsString(resultMap));  
 writer.flush();  
 writer.close();  
 }  
 *catch* ( IOException ex )  
 {  
 *throw new* RuntimeException(ex);  
 }  
}