

## 摘要

随着人工智能时代的到来，学习编程的热潮开始在全世界范围内兴起。2017年7月国务院颁发的《新一代人工智能发展规划》中指出：要开发基于大数据技术的在线教育平台，形成包括智能化学习、交互式学习的新型教育模式。如何利用好互联网技术的优势，打造一个以学习者为中心的智能在线编程教育平台，对推动教育信息化以及推广编程教育都有着极大的意义。本文的研究内容即使用先进的互联网信息技术为广大编程学习者提供免费优质的课程学习服务，以帮助学习者高质高效地在线学习编程知识。

本文在调研众多国内外的在线教育平台后，再结合编程学习者在线学习编程的需求，设计与实现了一个基于 Python 的专注于编程教育的在线教育学习平台。该平台基于典型的 B/S（浏览器/服务器）结构，网站后端采用广受欢迎、发展成熟、基于 Python 语言编写的 Web 框架 Django，数据库采用了关系型数据库 MySQL 存储数据，并使用 Redis 数据库缓存数据，前端技术主要有 Bootstrap 框架和 Ajax 异步技术。

通过进行需求分析，详细设计、系统实现、系统测试等软件开发过程后，本文采用快速迭代的方法开发出了该系统，并按照用户需求逐步完善，最后实现了在线教育平台的完整功能：注册登录、课程搜索、课程推荐、个人中心、课程上传、后台管理。本系统为学习者提供了搜课看课学课的在线学习服务，为教师提供了课程信息、课程视频和资料上传等功能，此外，还具有权限控制，日志管理等功能。

本文实现的在线教育平台具有良好的易用性和安全性，完成了一个在线教育平台基本的功能，方便了用户在线学习编程课程，总体而言达到了当初设计的目标。

**关键词：**MOOC，在线教育，Python，Django

## **Abstract**

With the advent of the era of artificial intelligence, the upsurge of learning programming has begun to rise all over the world. In the New Generation of Artificial Intelligence Development Plan issued by the State Council in July 2017, it is pointed out that the online education platform based on big data technology should be developed to form a new education mode including intelligent learning and interactive learning. How to make good use of the advantages of Internet technology and build a learner-centered intelligent online programming education platform is of great significance to promote education informatization and promote programming education. The content of this paper is to provide free and high-quality course learning service for the majority of programming learners by using advanced Internet information technology, so as to help learners learn programming knowledge online with high quality and efficiency.

After investigating many online education platforms at home and abroad, combined with the needs of programming learners online learning programming, this paper designs and implements a Python-based online education learning platform focusing on programming education. The platform is based on the typical B/S (browser/server) structure. The back end of the website adopts the popular, mature and Python-based Web framework Django. The database uses the relational database MySQL to store data, and uses Redis database to cache data. The front-end technologies mainly include Bootstrap framework and Ajax asynchronous technology.

After the software development process of requirement analysis, detailed design, system implementation and system testing, this paper develops the system by means of rapid iteration, and gradually improves it according to user needs. Finally, it realizes the complete functions of online education platform: registration, course search, course recommendation, personal center, course upload, background management. This system provides online learning service for learners to search for lessons and watch lessons. It also provides teachers with functions of course information, course video

and data upload. In addition, it also has functions of authority control and log management.

The online education platform realized in this paper has good usability and security, completes the basic functions of an online education platform, facilitates users to learn programming courses online, and generally achieves the original design goals.

**Keywords: MOOC, Online Education, Python, Django**

# 目录

摘要.....	I
ABSTRACT .....	II
第一章 绪论.....	1
1.1 引言 .....	1
1.2 研究现状 .....	2
1.3 研究目标和内容 .....	3
1.4 本文的组织结构 .....	4
第二章 系统相关技术.....	5
2.1 PYTHON.....	5
2.2 DJANGO.....	5
2.3 MYSQL.....	7
2.4 前端技术 .....	7
2.4.1 Bootstrap .....	7
2.4.2 Ajax .....	7
2.5 GIT 和 GITHUB .....	8
2.6 本章小结 .....	8
第三章 系统需求分析.....	9
3.1 功能性需求 .....	9
3.1.1 学习者功能需求分析.....	10
3.1.2 教师功能需求分析.....	11
3.1.3 管理员功能需求分析.....	11
3.2 非功能性需求 .....	12
3.3 本章小结 .....	12
第四章 系统设计.....	13
4.1 技术架构设计 .....	13
4.2 功能模块设计 .....	14
4.2.1 学习者模块.....	15
4.2.2 教师模块.....	19
4.2.3 管理员模块.....	19
4.3 数据库设计 .....	19
4.3.1 E-R 图设计 .....	20
4.3.2 数据表设计.....	23
4.4 本章小结 .....	25
第五章 系统实现.....	26
5.1 项目框架搭建 .....	26
5.1.1 开发环境.....	26
5.1.2 项目创建流程.....	26
5.1.3 项目目录说明.....	28

5.2	学习者模块实现 .....	29
5.2.1	注册登录功能实现 .....	29
5.2.2	课程搜索功能实现 .....	32
5.2.3	课程学习功能实现 .....	33
5.2.4	课程推荐功能实现 .....	39
5.2.5	个人中心功能实现 .....	41
5.3	教师模块实现 .....	42
5.4	后台管理模块实现 .....	44
5.5	本章小结 .....	45
<b>第六章</b>	<b>系统测试.....</b>	<b>46</b>
6.1	测试环境 .....	46
6.2	功能测试 .....	46
6.2.1	注册登录模块测试用例 .....	46
6.2.2	课程搜索模块测试用例 .....	47
6.2.3	课程学习模块测试用例 .....	47
6.2.4	课程推荐模块测试用例 .....	48
6.2.5	个人中心模块测试用例 .....	48
6.2.6	课程上传模块测试用例 .....	49
6.2.7	后台管理模块测试用例 .....	49
6.3	本章小结 .....	50
<b>第七章</b>	<b>总结与展望.....</b>	<b>51</b>
7.1	完成的工作 .....	51
7.2	存在的问题及下一步工作 .....	51
<b>参考文献</b>	<b>.....</b>	<b>52</b>
<b>致谢</b>	<b>.....</b>	<b>54</b>
<b>附录</b>	<b>.....</b>	<b>55</b>
附件 1	毕业设计文献综述 .....	55
附件 2	毕业设计开题报告 .....	55
附件 3	毕业设计外文翻译（中文译文与外文原文） .....	55

## 图目录

图 2-1 DJANGO 的 MTV 架构 <sup>[15]</sup> .....	6
图 3-1 在线教育平台整体功能模块.....	9
图 3-2 学习者需求用例 .....	10
图 3-3 教师需求用例 .....	11
图 3-4 管理员需求用例 .....	12
图 4-1 B/S 架构图 .....	13
图 4-2 在线教育平台架构图.....	14
图 4-3 注册模块业务流程图.....	15
图 4-4 登录模块业务流程图.....	16
图 4-5 课程搜索业务流程图.....	16
图 4-6 课程推荐业务流程图.....	18
图 4-7 用户 E-R 图 .....	20
图 4-8 课程 E-R 图 .....	20
图 4-9 课程视频 E-R 图 .....	21
图 4-10 课程章节 E-R 图.....	21
图 4-11 课程学习 E-R 图.....	21
图 4-12 课程评论 E-R 图.....	22
图 4-13 系统 E-R 图.....	22
图 5-1 项目的目录结构图 .....	28
图 5-2 注册页面 .....	29
图 5-3 激活邮件发送成功页面.....	31
图 5-4 系统首页 .....	32
图 5-5 课程搜索结果页 .....	33
图 5-6 课程详情页 .....	34
图 5-7 课程播放页 .....	35
图 5-8 课程 WIKI 页 .....	36
图 5-9 课程资料下载页 .....	37
图 5-10 评论发布页 .....	38
图 5-11 手记发布页 .....	39
图 5-12 课程推荐结果 .....	40
图 5-13 个人资料页 .....	41
图 5-14 我的课程页 .....	42
图 5-15 教师课程页 .....	43
图 5-16 教师上传视频.....	43
图 5-17 管理员添加用户权限 .....	44
图 5-18 管理员添加教师权限 .....	45

## 表目录

表 4-1 课程学习业务逻辑 .....	17
表 4-2 教师上传课程业务逻辑 .....	19
表 4-3 用户表 .....	23
表 4-4 课程表 .....	23
表 4-5 课程章节表 .....	24
表 4-6 课程视频表 .....	24
表 4-7 课程学习表 .....	24
表 4-8 课程评论表 .....	25
表 4-9 用户收藏表 .....	25
表 5-1 项目目录作用表 .....	28
表 6-1 注册登录测试用例表 .....	46
表 6-2 课程搜索测试用例表 .....	47
表 6-3 课程学习测试用例表 .....	47
表 6-4 课程推荐测试用例表 .....	48
表 6-5 个人中心测试用例表 .....	49
表 6-6 课程上传测试用例表 .....	49
表 6-7 后台管理测试用例表 .....	49

# 第一章 绪论

## 1.1 引言

随着科学技术的迅速发展，知识正以一种爆炸式的速度增长。身处于这个知识更迭速度飞快的现代社会中，人们逐渐发现学习和教育已经不是一劳永逸的事情。但在生活节奏快的现代社会中，人们只能利用工作之余或休息时的碎片化时间学习，现在传统的课堂教育往往会受到时间和空间上的限制，不能满足广大学习者的需求，而在线教育刚好满足了人们的这个需要。互联网技术的快速发展，使得在线学习成为了可能。在线教育相比于传统教育，不仅打破了时间及空间上的限制，方便了人们的学习，而且提供的资源之丰富，学习方式的多样化，是传统教育不可比拟的。近年来，随着“互联网+”的逐步深入与普及，传统教育行业正逐步与互联网实现深度融合<sup>[1]</sup>。尤其是在线教育具有不受时间和地点限制，碎片化学习等诸多优点，受到了广大学习者的欢迎。

人们对在线教育的强烈呼声催生了慕课这一伟大的模式，慕课，即 MOOC(Massive Open Online Course, 大规模在线开放课程)<sup>[2]</sup>，它是依托在线网络平台来进行授课的一种新的教学模式。慕课的出现，极大地推动了知识的传播，让世界上每一个热爱学习的人都能平等地获取这些开放的在线教育资源。通过 MOOC 在线平台，只要拥有一台可以连接上互联网的设备，任何人都可以很方便地学习到最新的课程资源。

自 MOOC 模式的在线教育平台从 2012 年在世界上流行以来，在线教育市场一直呈现着增长的趋势，而编程在线教育市场目前还拥有着广大的潜在用户人群，所以仍具有较大的发展潜力。同时，随着大数据时代与人工智能时代的到来，互联网企业的用人需求也在不断扩大，进入 IT 领域的毕业生也越来越多。为了帮助社会和企业培养合格和专业的互联网人才，也为了帮助有志从事 IT 技术工作的用户以及学生群体提升技术能力，迫切需要建设一个满足编程学习者实际需要的



在线编程教育平台。

信息化教育以其跨地域、跨时空的优势，最大限度地满足了人们随时随地学习知识的要求，以其自身巨大的潜力丰富着教育内容和教育方式<sup>[1]</sup>。将来的在线教育平台需要引入大数据和人工智能技术，针对不同的受众来提供个性化、智能化的优质高效的课程教学。

## 1.2 研究现状

随着互联网技术的进步，互联网与教育快速融合，极大地颠覆了传统教育的概念。MOOC是在线教育的一种新模式，起源于基于互联网的开放课程，最早可追溯到2007年<sup>[3]</sup>。在2012年年初时，MOOC由哈佛、麻省理工学院等国际名校引领，并于短时间内快速席卷全球<sup>[4]</sup>。只需拥有一台可以连接上互联网的设备，通过这些在线平台，任何人都可以很方便地获取到这些只有极少数考进这些大学的人才能享受到的学习资源。

Coursera、edX和Udacity是国外最受欢迎的三大慕课平台。其中，Udacity专注于职业技能教育领域，该平台提供的课程多与计算机科学、互联网技术、数学等相关，这主要是因为Udacity 课程的设计与录制是由 Google, Facebook, Twitter 等互联网公司负责的<sup>[5]</sup>。与 Coursera、edX 不同的是，它的课程视频与家庭作业之间并未分离，而是有一个完整的学习路线，如果完成了才能进入下一单元的学习<sup>[6]</sup>；此外，Udacity 拥有一个非常强大的 wiki 平台，每门课程都有专属的 wiki 链接，能链接至课程各个单元，查看相应单元的核心知识点、课堂截屏、重点笔记、作业练习等，这个独特的设计理念，不仅可以解决学生在学习课件或视频过程中产生的一些问题，即便学生没有时间观看课程教学视频，也能通过 wiki 这种途径来进行自学<sup>[7]</sup>。

若2012年被称之为慕课的元年<sup>[8]</sup>，那么2013年则被称为中国的“慕课元年”。以网易公开课、学堂在线、中国大学MOOC、慕课网为代表的在线教育平台迅速在中国兴起，并受到了大量互联网用户的欢迎。国内在线教育市场的前景被一致看好，BAT等互联网巨头也开始迅速入局在线教育，比如百度的“百度传课”与“作

业帮”，阿里巴巴的“淘宝教育”，腾讯的“腾讯课堂”。

目前，我国现有的MOOC 平台中专注于IT职业技能教育主要有慕课网、极客学院、尚硅谷、51CTO学院和传播智客，该类平台并没有依赖顶尖大学的教学资源，而是通过邀请信息技术和互联网领域的资深工程师来策划和录制课程，许多课程侧重于一项技能、一个产品案例或一个具体的问题解决方案<sup>[9]</sup>。此类平台经过几年的快速发展，凭借其免费而优质的课程赢得了广大用户的口碑，现在正逐渐向知识付费的模式过渡。以慕课网为例，该平台自2017年推出了收费的实战课程与就业路径系列，其目的是加强课程的实践性，让学生在项目实战中提升自身的技能，进而提高就业竞争力。然而，目前国内的专业技能教育平台与国外仍有相当大的差距，比如知识体系不全、课程质量参差不齐、课程完成率不高、知识付费体系仍未完善等。

目前，国内慕课市场的规模仍在不断扩大。中商产业研究院在2018年发布的《中国在线教育行业市场前景研究报告》提供了国内在线教育规模及融资的数据：2017年用户规模达到1.55亿人，预计2018年将突破1.79亿人，市场规模有望突破3000亿元<sup>[10]</sup>，从细分领域上来看，K12/职业培训/语言学习一直是最热的投资领域。

### 1.3 研究目标和内容

本文的研究目标是利用 Python、Django 等技术来开发和实现一个功能相对完整的在线编程教育平台，以推动编程知识的传播和分享，让每一个热爱编程的人都能享受到免费而优质的课程。

本文的研究内容主要如下：

1. 对在线教育平台的关键技术做调研，学习 Python、Django 的基础知识。
2. 在需求分析阶段，详细调研了在线教育平台的需求。
3. 在系统设计阶段，设计系统的技术架构，并根据用户的需求给出了每个功能模块的详细设计以及数据库的设计。
4. 在系统实现阶段，完成在线教育平台的前端及后端的编码。

5. 在系统测试阶段，使用软件测试的理论和方法对系统进行测试。

本系统是本文作者结合软件工程理论知识所进行的一次深入实践，从需求分析、详细设计，再到编码和实现，处处都体现着软件工程的思想。

## 1.4 本文的组织结构

本文主要分为七个章节展开论述，以下将对每个章节进行简要的介绍。

第一章，绪论，主要分析了在线教育平台的研究背景，阐述当前国内外研究现状和意义，并指出研究的目标和内容，最后对本文的每个章节做概要介绍。

第二章，系统相关技术，简要介绍了本系统使用到的关键技术，如 Python、Django、MySQL 等。

第三章，系统需求分析，主要分析了在线教育平台的功能性需求，以及确定了一些非功能性需求。

第四章，系统详细设计，分作系统技术架构、功能模块和数据库三大部分展开描述。

第五章，系统实现，根据系统设计开发每个功能模块，并对每个功能模块的实现方法进行详尽的介绍与说明。

第六章，系统测试，根据需求说明书，对系统进行功能测试。

第七章，总结与展望，本章总结了项目开发所做的工作，并分析了本系统存在的不足之处，最后提出了改进的思路。

## 第二章 系统相关技术

本章主要介绍了本系统在项目开发过程中所使用到的关键技术，包括编程语言 Python、Python web 框架 Django、数据库 MySQL、前端技术 Bootstrap 及 Ajax、项目管理工具 Git 与 GitHub。下面将从背景、优缺点、应用等方面来对相关技术展开介绍。

### 2.1 Python

Python 是一种简单的、解释型的、交互式的、可移植的、面向对象的高级语言<sup>[11]</sup>。Python 具有语法简单、功能强大、移植性好等诸多优点，受到了广大开发者的欢迎。Python 还拥有大量丰富的第三方库，使用这些第三方库可以极大地提高开发的效率。在 2019 年 6 月的 TIOBE 编程语言排行榜中，Python 排名第四，仅次于 Java 和 C 语言<sup>[12]</sup>。

现在无论 PC 端还是移动互联网的 Web 开发工作，对产品做的各种尝试都需要更快地拿出模型并进行迭代，创业公司尤甚。而 Python 很适合进行产品的快速迭代开发，所以对于初创公司或规模较小的开发团队而言，使用 Python 作为 Web 开发的编程语言是一种不错的选择。

### 2.2 Django

Django 于 2005 年 7 月诞生，经过十多年的发展，已经成为了当下最受欢迎的开源 Python web 框架，该框架内置多种 Web 开发组件，同时还集成了一个轻量级 Web Server，使开发人员可以最少的代码、最快的速度、方便高效地进行各种 Web 应用的开发和调试<sup>[13]</sup>。目前，Django 已经在 Web 开发中被广泛使用，国内外知名的互联网公司如 Instagram、豆瓣都在使用 Django 来开发它们的网站。

Django 使用的架构是类似于 MVC 的 MTV 架构<sup>[14]</sup>，由 Model(模型)、Template

（模板）、View（视图）三者搭配组成。其中 M 代表的是 Model，即数据存取层，这一层主要进行数据模型相关的定义与处理，完成数据的增删改查。T 代表的是 Template，即模板层，这一层通过接收视图层传递过来的数据，使得网站能够显示动态的数据。V 代表 View，即视图层，该层主要用来处理业务逻辑。下图 2-1 即 Django 使用到的 MTV 架构。

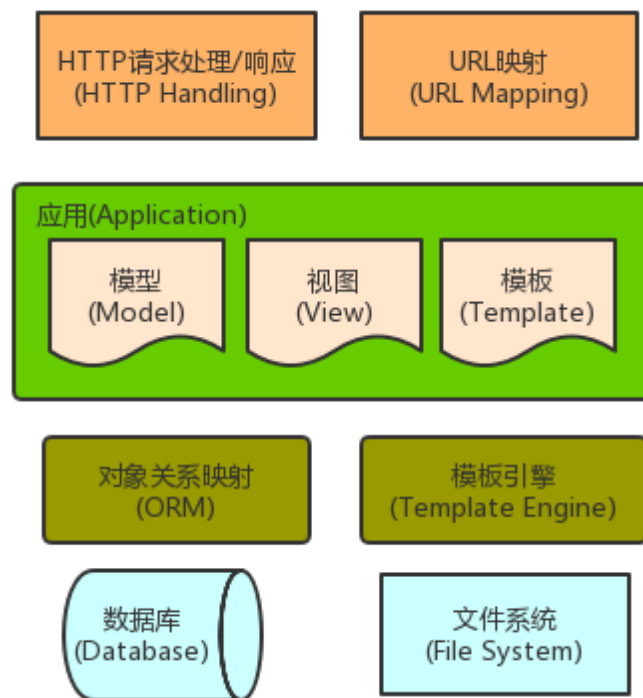


图2-1 Django的MTV架构<sup>[15]</sup>

如上图所示，Django 除了拥有 Model（模型）、Template（模板）、View（视图）三个 MTV 架构的组成部分之外，还有 URL 映射（URL Mapping）和对象关系映射（ORM）两个重要的组件。下面将简要介绍这两个组件：

对象关系映射（ORM），全称为 Object Relational Mapping<sup>[16]</sup>。ORM 能将数据库中的关系型数据映射成对象，允许应用程序以面向对象的方式来操作数据库。Django 在底层封装了这些数据库 SQL 操作，并提供了操作这些数据的 API 接口，这样开发者只需要使用这些 API 接口，就可以直接操作数据对象，不必再编写 SQL 语句与数据库交互。

URL 映射（URL Mapping）是 Django 中的路由分发组件，通过 URLconf（即

路由配置系统）来进行配置。简单而言，URLconf 提供了 URL 与视图函数之间的映射关系，是 Django 访问网站时的入口。当用户要访问某个 URL 时，URLconf 会对这个 URL 进行正则表达式匹配，然后再决定执行相应的视图函数。

## 2.3 MySQL

MySQL 是当前主流的开源数据库管理系统，已被甲骨文公司于 2009 年收购。MySQL 以其性能好、体积小、操作简便、使用成本低等优势<sup>[17]</sup>，受到了广大开发者的青睐。目前，许多大型的互联网公司都在使用 MySQL，比如谷歌、脸书、阿里巴巴和腾讯等。MySQL 现在对外发布两个版本：免费开源版和商业收费版，不过中小型网站及个人网站的数据并发处理量较小，所以免费版已经能完全满足需要了。

## 2.4 前端技术

### 2.4.1 Bootstrap

Bootstrap 是一个流行的开源前端框架，由 Twitter 的工程师开发，使用 Bootstrap 可以开发出 Twitter 风格的前端界面。

Bootstrap 之所以受到广大开发者的喜爱，与以下特性是密不可分的：

1. Bootstrap 内置了许多丰富的 CSS 样式及组件，避免了前端代码的重复编写
2. Bootstrap 提供了一个响应式的栅栏式布局，可以适配移动端
3. Bootstrap 基于 HTML、CSS 和 JavaScript 语言编写，扩展性强

Bootstrap 简单易用，文档齐全。使用 Bootstrap，可以让后端开发工程师也能快速搭建出美观的 UI 界面。

### 2.4.2 Ajax

Ajax，英文全称为 Asynchronous JavaScript and XML。从 Asynchronous 即可得知 Ajax 其实是一种异步操作，这意味着当网页内容动态更新的时候，只需要刷

新指定区域的网页而不必重新加载整个网页，这样既降低了服务器资源的损耗，也提升了前端页面的响应速度。Ajax 实现的原理就是在后台服务器和前台页面之间用了一个中间件，让用户在页面上的操作可以和后台的服务器响应分开，这样就实现了异步<sup>[18]</sup>。

## 2.5 Git 和 GitHub

Git 是一个分布式的开源版本管理系统，使用 Git 可以很方便地进行多人协作和代码管理。开发团队可以通过 Git 来记录代码提交或修改的历史，回滚到某个特定的版本。

GitHub 是为开发者提供 Git 仓库的托管服务<sup>[19]</sup>。GitHub 还为开发者提供了一个图形化界面，方便了用户使用 Git 服务进行代码共享，除此之外，GitHub 还提供了 Pull Request、Issue、Wiki 等功能，让开发者能更好地协作开发软件。

## 2.6 本章小结

本章先介绍了项目的编程语言 Python，然后对 Django 框架的架构及组件做了简要的描述，然后从技术产生背景，优缺点等方面依次介绍了 MySQL、Bootstrap、Ajax、Git、GitHub 等技术。

## 第三章 系统需求分析

需求分析是软件开发的第一个阶段，也是至关重要的一环。在需求分析阶段往往需要对概念进行研究和细化，需求分析得越透彻，在后期的设计与开发中需要改动的工作就越少。此外，需求分析中很重要的一点是，需要向用户提取需求，只有知道了用户真正需要什么，才能开发出能使客户满意的软件。本文的需求分析主要分成两个部分：功能性需求分析和非功能性需求分析。

### 3.1 功能性需求

在线教育平台旨在为编程学习者提供一个免费的在线编程学习平台。本系统的用户类型主要有三种：学习者，教师，管理员，针对以上用户的需求进行详细的分析后，设计出了在线教育平台的整体的功能模块图，如图 3-1 所示。

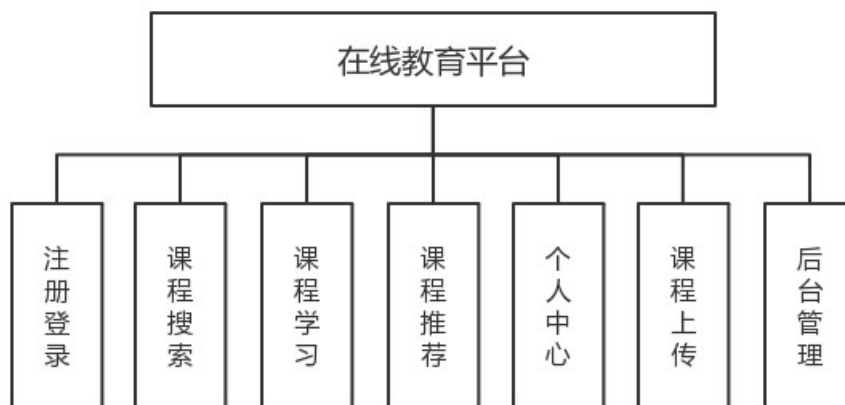


图3-1 在线教育平台整体功能模块

下面将给出以上三类用户，即学习者，教师，管理员的功能需求分析。



### 3.1.1 学习者功能需求分析

学习者在注册之前可以浏览网站首页，以及课程的基本信息，当用户想要学习课程时，系统会自动提示用户进行登录或注册操作。学习者需要填写注册邮箱和密码等相关信息以完成注册。用户可输入刚注册的账号信息进行登录，登录成功后，用户可以开始学习课程，比如观看课程视频，学习课程 wiki 文档，参与课程讨论，下载课程资料文件，记录课程笔记。用户还可以收藏课程，关注教师，进入个人中心修改个人信息、查看学习的课程以及收藏的课程和教师。

总结以上的具体需求，学习者功能需求可以用以下的用例图 3-2 来表示：

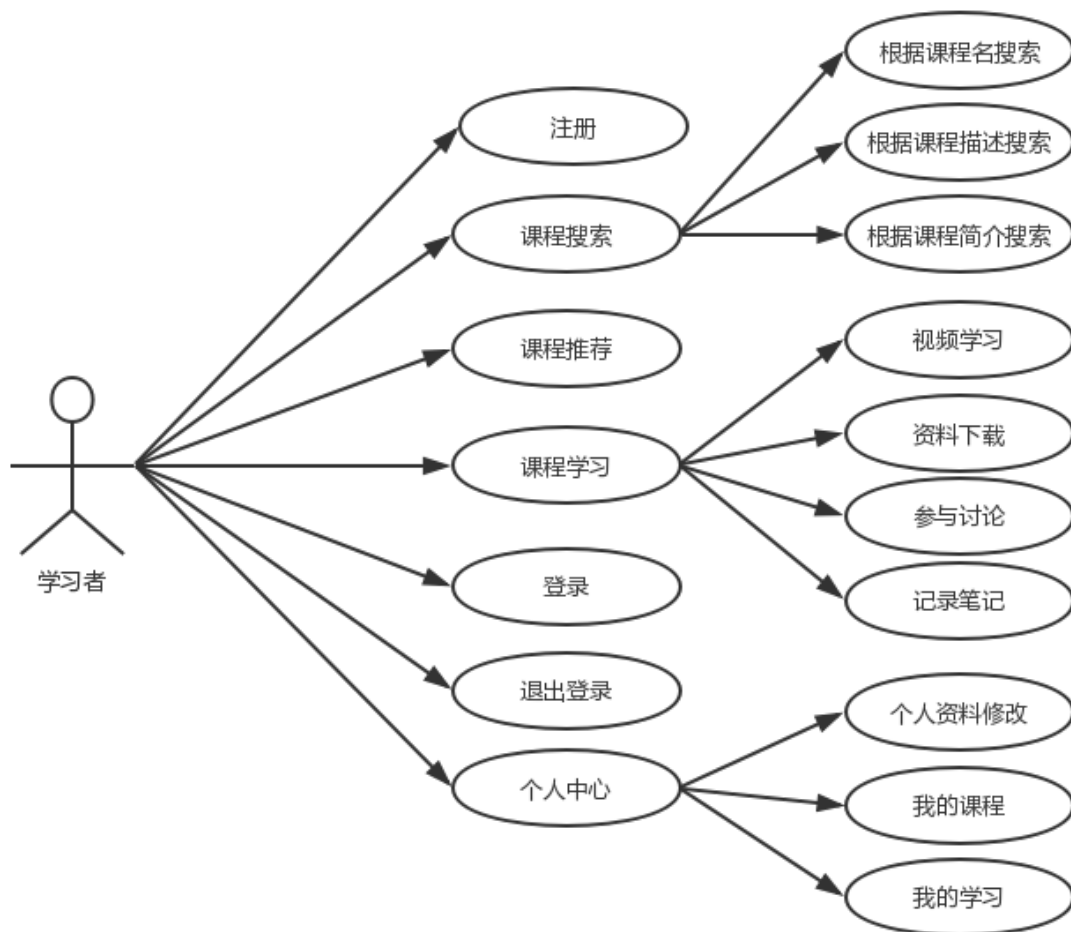


图 3-1 学习者需求用例

### 3.1.2 教师功能需求分析

教师是一种特殊的用户类型，是在学习者的基础上加上一定的权限转化而成的。教师除了拥有普通学习用户的全部权限外，还可以上传课程的基本信息和资源，比如上传课程视频、课程资料、wiki 文档、课程章节信息等，教师还可以对自己所教授的课程的基本信息进行管理。

教师的功能需求可以用以下用例图 3-3 表示：

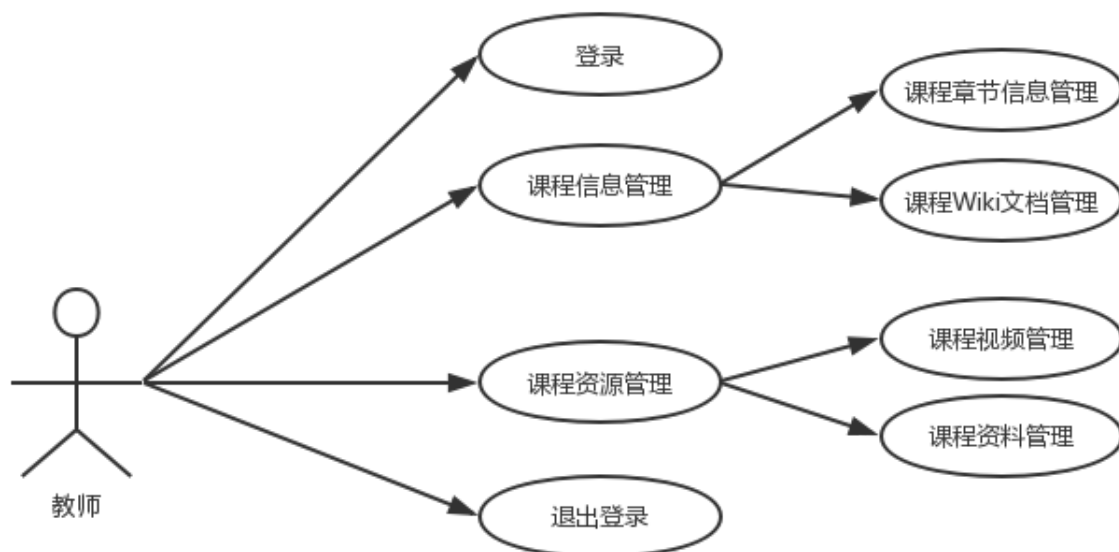


图3-3 教师需求用例

### 3.1.3 管理员功能需求分析

管理员拥有网站的所有权限，是维护网站正常健康运行的管理者。管理员可以对用户进行管理，添加或删除用户，也可以管理课程，比如审核教师上传的课程相关信息。除此之外，管理员还可以进行权限管理，为用户分配教师的权限，管理员还可以查看管理操作的日志记录，以便管理操作失误后可以及时恢复数据。管理员的功能性分析可以用以下用例图 3-4 表示：

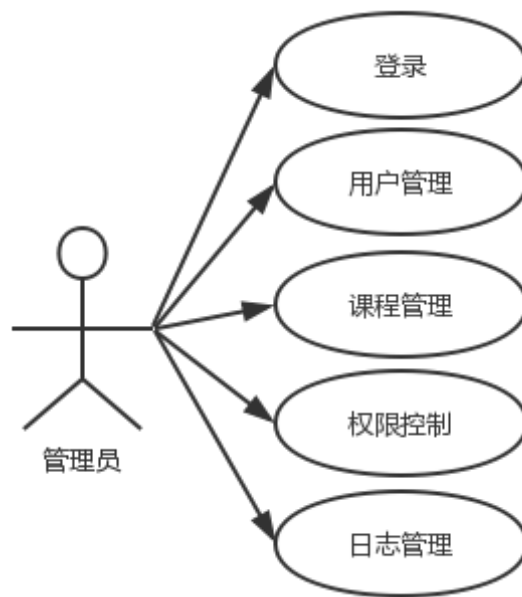


图3-4 管理员需求用例

## 3.2 非功能性需求

完成以上的功能性需求还不能完全满足用户的需求，除此之外，还要满足一些非功能性需求。

1. 可靠性。网站要拥有较好的可靠性，故障率低。在线运转时间长
2. 正确性。网页正确显示，数据同步正常
3. 安全性。需要保证用户信息的安全性，避免泄露或受到篡改。
4. 交互性好。网页响应时间要尽量短，一般要控制在 3 秒之内响应。平台的界面设计要符合人的一般审美标准，简洁美观，以提升用户的使用体验。

## 3.3 本章小结

本章分析了在线教育平台的功能性需求，绘制了系统的整体功能结构图，并将用户的功能性需求细化，给出了学习者、教师与管理员的用例图，最后还对一些非功能性需求做出了规定。

## 第四章 系统设计

本章分别从技术架构、功能模块、数据库三个方面对系统进行详细的设计，给出了系统的整体技术架构图，划分并设计每个功能模块，最后给出了系统数据库的 E-R 图与数据表设计。

### 4.1 技术架构设计

本平台属于典型的 B/S 架构，B/S(Browser/Server)结构即浏览器和服务器结构<sup>[20]</sup>，如图 4-1 所示。根据 B/S 架构的思想，可以将平台的架构大体分成三大部分：浏览器端、服务器端、数据库。

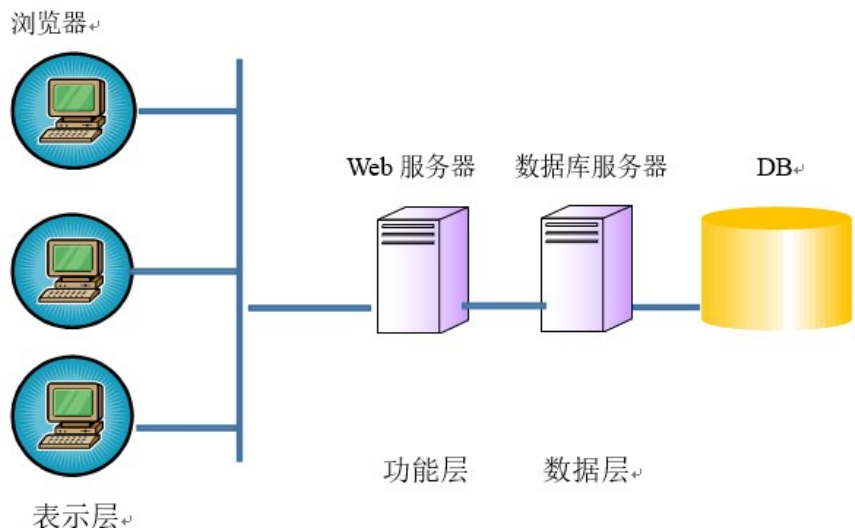


图4-1 B/S架构图

浏览器端：当用户在浏览器端浏览网页时，浏览器端先向网站服务器发送 HTTP 请求，然后接收网站服务器发回的 HTTP 响应，最后使用 HTML、CSS 以及 JavaScript 等前端技术构建网页。

服务器端：服务器端收到用户请求后，有两种不同的处理方式处理请求，若为静态请求，则由服务器端直接处理并将结果返回客户端；若为动态请求，则交给 Django 处理，Django 实现相应的业务逻辑处理后，将处理结果返回服务器端，

再由服务器端传至浏览器端。

数据库：数据库主要用来存储网站的数据。服务器端通过连接数据库，即可完成对数据的存取操作。

基于 B/S 架构的设计特点，并结合 Django 框架的 MTV 架构思想与平台的具体业务需求，可以对以上架构进行改进。在线教育平台的技术架构设计如图 4-2 所示。

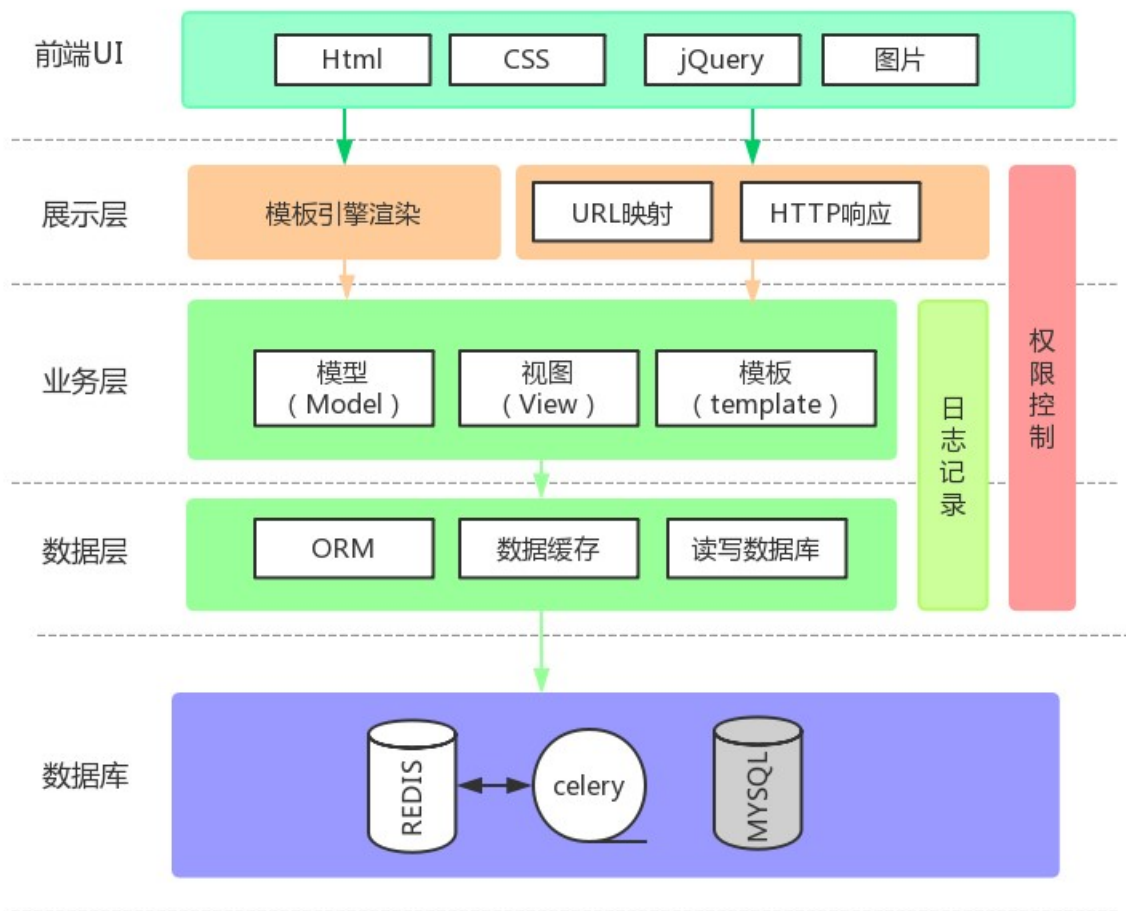


图4-2 在线教育平台架构图

## 4.2 功能模块设计

进行需求分析后，可以将平台的功能按照用户的类型分为三个功能模块：学

习者模块，教师模块和管理员模块。下面将对以上功能模块的设计做详尽的介绍。

### 4.2.1 学习者模块

学习者模块主要分为注册、登录、课程搜索、课程学习、课程推荐、个人中心等功能。

#### (1) 注册

在线教育平台的注册模块只有学生用户的注册，教师用户账号由管理员分配。学生用户需要输入有效的邮箱地址、用户名及密码来进行注册操作。注册完成后，还需要进行邮箱的激活，只有激活后才能成功登录平台。用户注册的业务流程如图 4-3 所示：

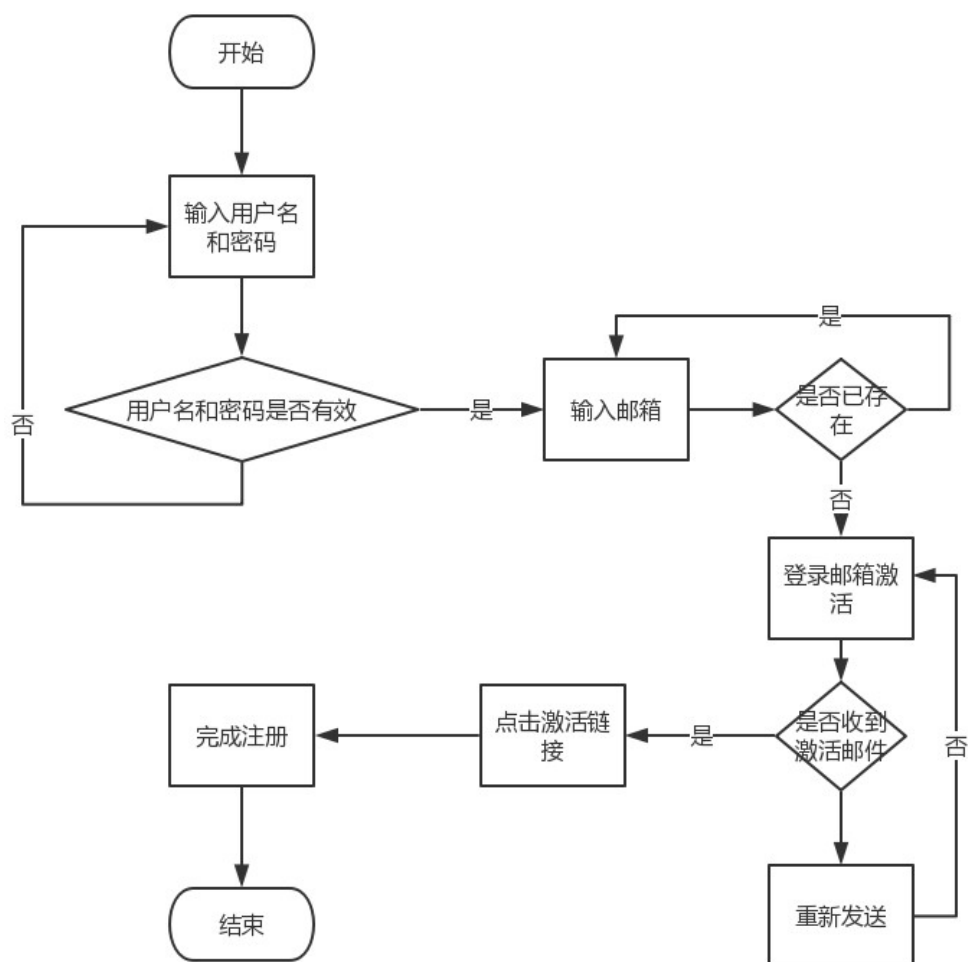


图4-3 注册模块业务流程图

## (2) 登录

用户可以输入用户名和密码进行登录。为了保证整个在线教育平台的安全性，平台对用户的登录权限做了严格的限制，这便意味着普通的学生用户不能登入平台的后台管理系统，只能在网站前台登录。

教师与管理员是在网站的后台管理系统进行登录的，其登录业务逻辑与学生用户的一致，在此不再赘述。

## (3) 课程搜索

课程搜索模块是为了方便用户寻找课程而设计的。用户通过在搜索栏输入关键字，然后根据用户输入的关键字在数据库中对课程名、课程简介、课程描述做模糊匹配，最后返回符合的结果。用户登录的业务流程如图 4-4 所示，课程搜索的业务流程如图 4-5 所示。

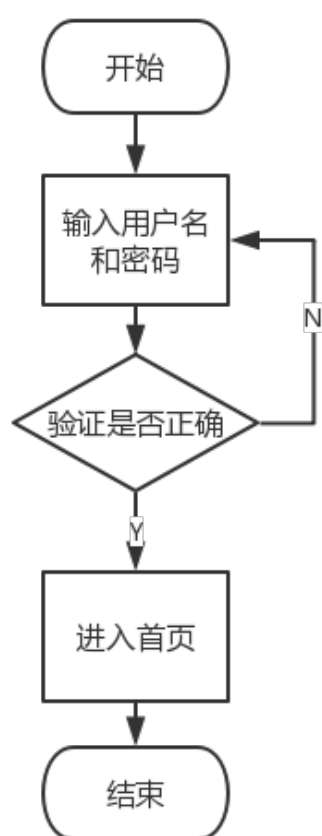


图4-4 登录模块业务流程图

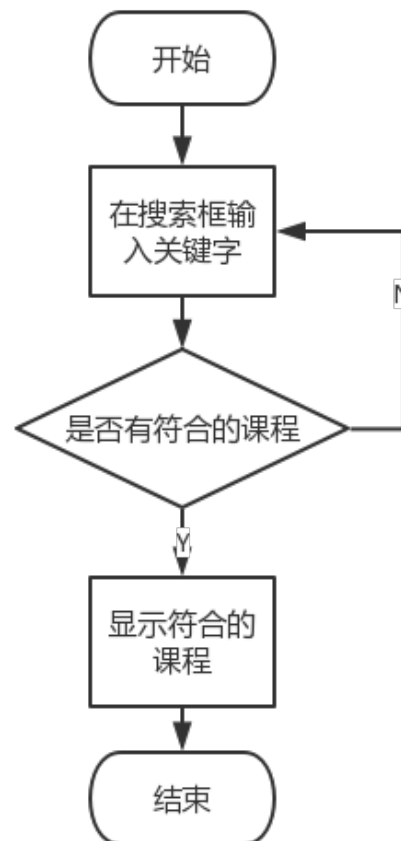


图4-5 课程搜索业务流程图

#### (4) 课程学习

课程学习功能是整个在线教育平台最核心的部分，主要分为课程信息查看、课程视频观看、课程评论发表、课程笔记发布、课程 Wiki 阅读、课程资料下载。课程学习操作的业务逻辑如表 4-1 所示：

表 4-1 课程学习业务逻辑

编号	所在页面	操作逻辑
1	课程详情页	查看课程基本信息
2	课程详情页	学习课程 wiki
3	课程详情页	课程资料下载
4	课程详情页	点击视频，进入视频播放页
5	视频播放页	观看视频
6	视频播放页	发表评论
7	视频播放页	发布手记

#### (5) 课程推荐

课程推荐功能设计的目的是让用户能找到自己感兴趣的课程，同时推荐平台上的一些优质课程。本文设计了一个简单的课程推荐算法，即对新注册用户而言，则推荐平台上学习人数多的热门课程，而对已经有学习记录的老用户，则根据其学习过的课程来推荐一些具有相同课程标签的课程。课程推荐模块的业务流程如图 4-6 所示：



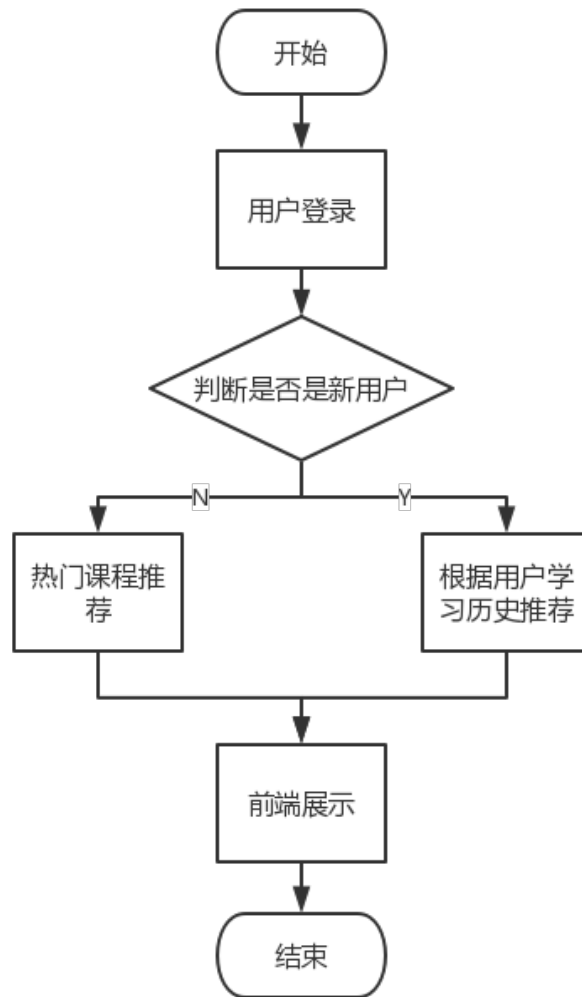


图4-6 课程推荐业务流程图

## (6) 个人中心

在个人中心模块，用户可以查看并修改个人信息、查看已学习和收藏的课程，方便继续进行课程的学习。下面是个人中心模块的详细设计：

个人信息查看，编辑：用户可以在个人中心页查看自己的个人资料，也可以修改自己的个人信息，当新填写的信息有效时，修改成功。

我的课程：用户可以查看之前学习过的课程，并点击课程继续学习

我的收藏：用户可以查看已收藏的课程，取消收藏课程，也可点击课程继续学习。

### 4.2.2 教师模块

教师模块是专门为教师上传课程基本信息和资源文件而设计的，教师可以管理课程章节信息、课程 Wiki，也可以上传课程视频、课程资料等资源文件。该模块引入了权限控制，教师只能管理其教授的课程，对其它课程只能浏览而不能修改。教师进行课程上传的业务逻辑如表 4-2 所示：

表 4-2 教师上传课程业务逻辑

编号	操作对象	操作逻辑
1	课程章节信息	查询、添加、修改、删除
2	课程 Wiki	使用 Markdown 编辑器编写、修改 Wiki
3	课程视频	查询、添加、修改、删除
4	课程资料	上传、修改、删除

### 4.2.3 管理员模块

管理员模块主要分为用户管理、课程管理、权限控制、日志管理等功能。下面简要介绍一下以上列出的功能。

用户管理：包括用户信息的查询、添加、修改、删除

课程管理：课程信息的查询、添加、修改、删除

权限控制：设置教师组权限，为教师用户设置教师组权限

日志管理：系统日志查询、添加、删除、导出

## 4.3 数据库设计

数据库设计是软件项目开发中极为重要的一环，其设计的好坏则会直接影响到网站数据库系统的效率和质量。本文采用 MySQL 关系型数据库作为数据库管理系统，并根据用户的需求设计了合理的数据模型，使用 Navicat 生成了相应的数据表。

本系统主要设计了 12 张数据表，分别为用户表、课程表、课程一级分类表、课程二级分类表、课程章节表、课程视频表、课程 Wiki 表、课程手记表、课程评论表、课程学习表、用户收藏表、邮箱验证码表。下面将对这些数据表的概念

结构设计以及逻辑结构设计做出详尽的介绍。

### 4.3.1 E-R 图设计

这一小节列出了在线教育平台的重要的 E-R 图以及系统的 E-R 图。

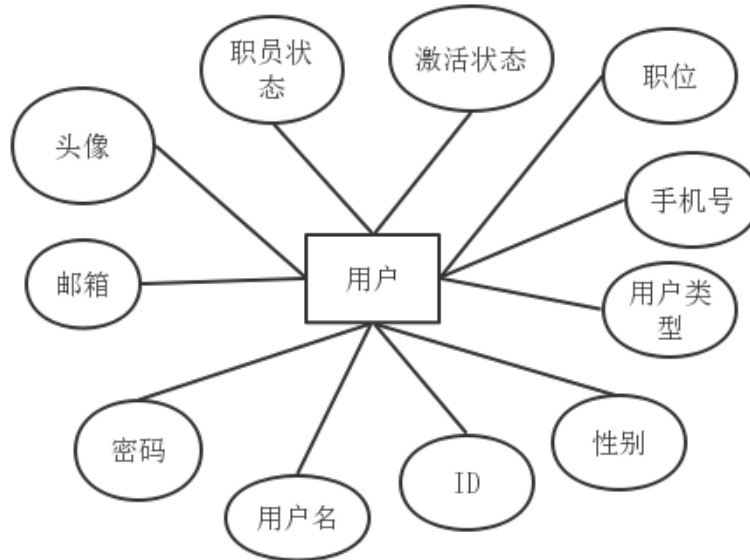


图4-7 用户E-R图

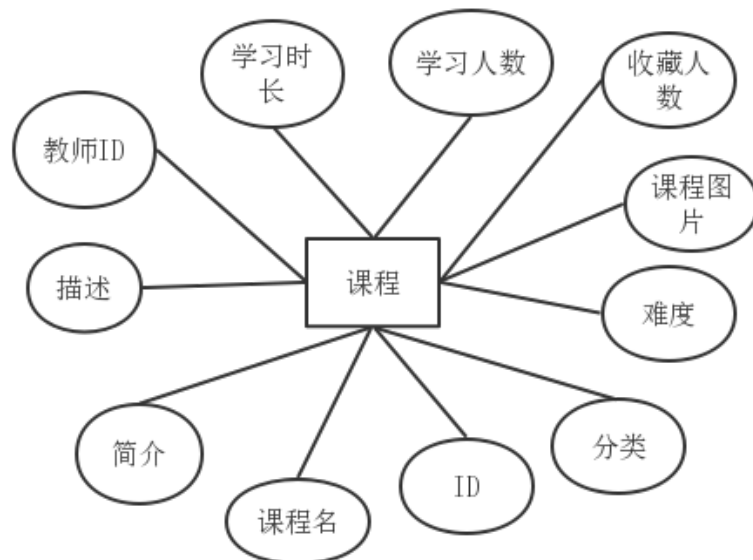


图4-8 课程E-R图

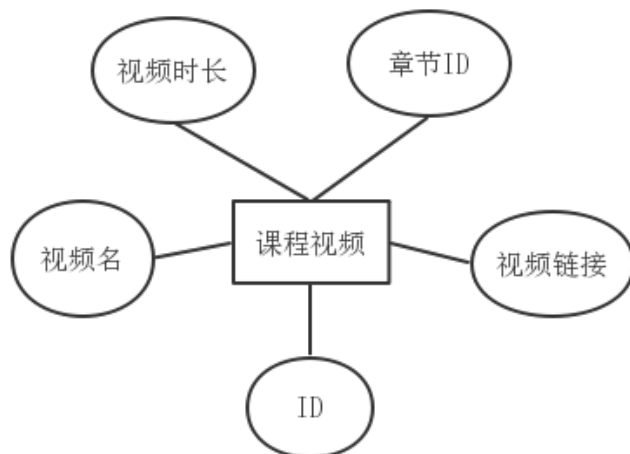


图4-9 课程视频E-R图

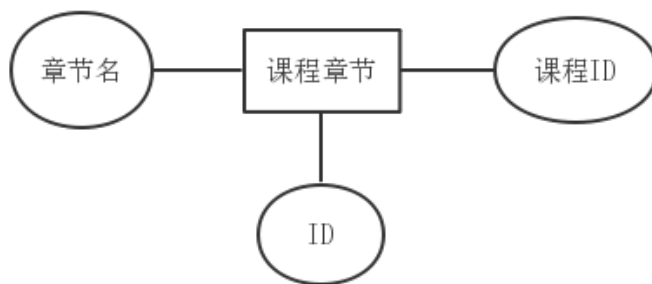


图4-10 课程章节E-R图

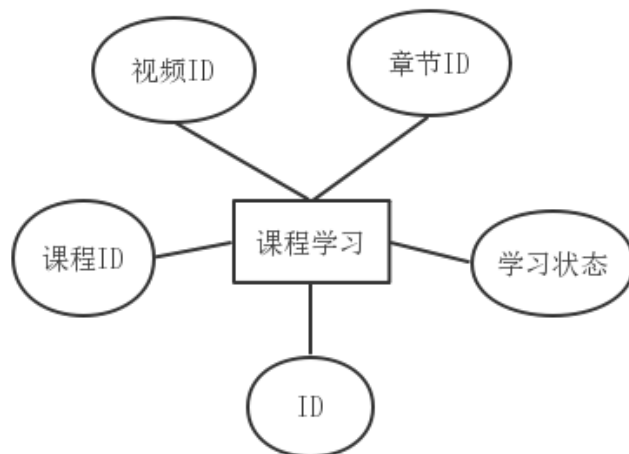


图4-11 课程学习E-R图

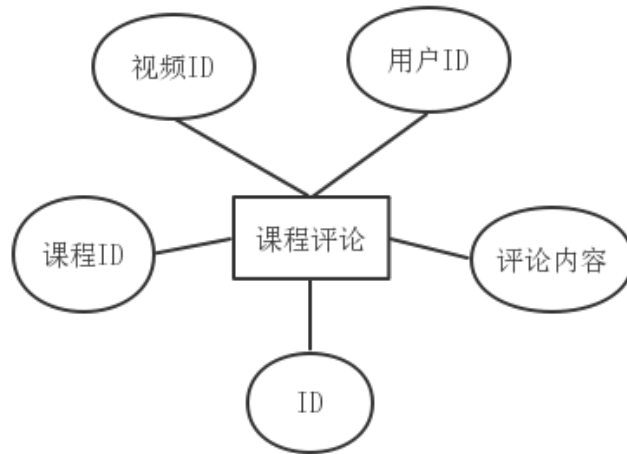


图4-12 课程评论E-R图

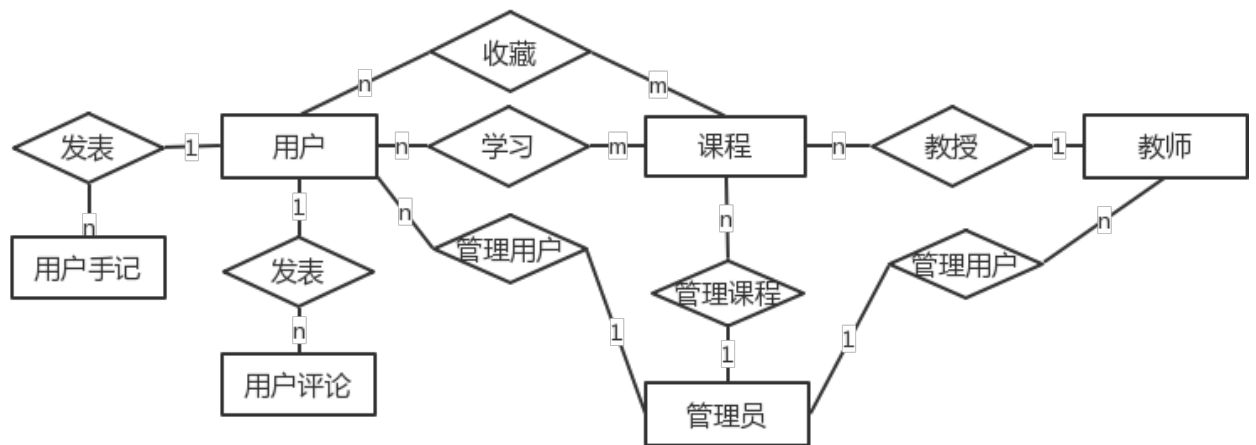


图4-13 系统E-R图

### 4.3.2 数据表设计

1. users\_userprofile: 用户表, 保存用户信息。具体的表结构设计见表 4-3:

表 4-3 用户表

字段名	字段类型	字段含义	备注
id	bigint	ID	主键
nick_name	varchar	用户名	唯一
pass_word	varchar	密码	
email	varchar	邮箱	
user_image	varchar	头像	
job	varchar	职位	
mobile	varchar	手机号	
user_type	tinyint	用户类型	(0: 管理员, 1: 教师, 2: 学生)
gender	varchar	性别	
signature	varchar	签名	
is_active	tinyint	激活状态	(0: 未激活, 1: 已激活)
is_staff	tinyint	职员状态	(0: 非职员, 2: 职员)
is_superuser	tinyint	超级管理员状态	(0: 否, 1: 是)
last_login	datetime	上次登录时间	
date_joined	datetime	注册时间	

2. courses\_course: 课程表, 用来存储平台中的课程信息。具体的表结构设计见 4-4:

表 4-4 课程表

字段名	字段类型	字段含义	备注
id	bigint	ID	主键
name	varchar	课程名	唯一
desc	varchar	课程简介	
detail	varchar	课程描述	
image	varchar	课程缩略图	
degree	varchar	课程难度	(cj: 初级, zj: 中级, gj: 高级)
learn_times	int	课程时长	
stu_times	int	学习人数	
fav_times	int	收藏人数	
teacher_id	int	教师	教师 ID, 外键
cl_root_id	int	一级分类	一级分类 ID, 外键
cl_detail_id	int	二级分类	二级分类 ID, 外键
you_need_know	varchar	课程须知	
teacher_tell	varchar	教师建议	

字段名	字段类型	字段含义	备注
add_time	datetime	添加时间	

3. **courses\_lesson**: 课程章节表, 存放课程章节的相关信息。具体的表结构设计见表 4-5:

表 4-5 课程章节表

字段名	字段类型	字段含义	备注
id	bigint	ID	主键
name	varchar	章节名	
course_id	varchar	课程 id	外键
add_time	datetime	添加时间	

4. **courses\_video**: 课程视频表, 存放课程视频的相关信息。具体的表结构设计见表 4-6:

表 4-6 课程视频表

字段名	字段类型	字段含义	备注
id	bigint	ID	主键
name	varchar	视频名	
video_file	varchar	视频链接	
learn_times	int	视频时长	
lesson_id	int	课程章节 id	外键
add_time	datetime	添加时间	

5. **operation\_usercourse**: 课程学习表, 存储用户学习过的课程。具体的表结构设计见表 4-7:

表 4-7 课程学习表

字段名	字段类型	字段含义	备注
id	bigint	ID	主键
user_id	int	用户 ID	外键
course_id	int	课程 ID	外键
video_id	int	视频 ID	外键
add_time	datetime	添加时间	

6. **operation\_coursecomments**: 课程评论表, 存储用户在课程或课程视频下发

表的评论。具体的表结构设计见表 4-8:

表 4-8 课程评论表

字段名	字段类型	字段含义	备注
id	bigint	ID	主键
comments	varchar	评论内容	
user_id	int	用户 ID	外键
course_id	int	课程 ID	外键
video_id	int	视频 ID	外键
add_time	datetime	添加时间	

7. operation\_userfavourite: 用户收藏表, 记录用户收藏的课程。具体的表结构设计见表 4-9:

表 4-9 用户收藏表

字段名	字段类型	字段含义	备注
id	bigint	ID	主键
fav_id	int	收藏 ID	
fav_type	int	收藏类型	(1: 课程, 2: 教师)
user_id	int	用户 ID	外键
add_time	datetime	添加时间	

除了以上列出的数据表外, 还有 Django 自身生成的数据表, 比如权限表、权限组表、日志记录表等, 由于本文篇幅有限, 所以暂不描述。

值得说明的是, 诸如图片和视频等资源文件并不直接保存在数据库中, 而是保存在专门的对象存储服务器上, 比如本地磁盘, 这样只要在数据库中保存这些文件的链接, 就能方便地访问到这些资源。

## 4.4 本章小结

本章分别从技术架构、功能模块、数据库三个方面对系统进行详细的设计, 给出了系统的整体技术架构图, 划分并设计每个功能模块, 最后给出了系统数据库的 E-R 图与数据表设计。



## 第五章 系统实现

本章先介绍基于 Django 的项目框架搭建的一般过程，再从使用者的角度分别给出每个功能模块的具体实现，其中，使用者分为学习者、教师、管理员三种类型。

### 5.1 项目框架搭建

#### 5.1.1 开发环境

本平台在 Windows 10 操作系统上开发完成，开发主要使用的工具和环境如下：

- 1) Python 3.6
- 2) Django 2.1.7
- 3) MySQL 5.6
- 4) PyCharm 2018.3.5
- 5) Bootstrap 3.3

#### 5.1.2 项目创建流程

Django 初始化 Web 项目十分方便，只需要简单的几行命令便可完成。下面将逐步介绍一下创建本项目的过程。

##### 1. 新建项目

进入项目开发目录，并在命令行中输入：

```
$ django-admin startproject MoocOnline
```

此时会在当前目录生成一个名为 MoocOnline 的新目录，这个目录就是项目的根目录。

##### 2. 新建应用

本系统采用了模块化开发的思想，即将整个项目分成几个模块，并在每个模

块中编写相应的代码，这样便降低了模块间的耦合性，方便了项目的维护，系统在添加新功能时也不会影响到之前的代码。

Django 也为开发者提供了这种模块化开发的功能，可以通过新建应用 **app** 的方式创建一个模块，下面是 Django 新建应用的命令：

```
$ python manage.py startapp users
```

通过执行以上的命令，一个名为 **users** 的新应用就创建好了。

### 3. 设置 settings.py 文件

settings.py 文件是项目中的全局设置文件，项目数据库的相关配置也是写在该文件中的。下面是在 settings.py 文件中连接 MySQL 数据库的相关配置信息：

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': "mooc-online",  
        'USER': "root",  
        'PASSWORD': "123456",  
        'HOST': "127.0.0.1"  
    }  
}
```

### 4. 数据库迁移与同步

Django 通过 models.py 中的模型类与数据库产生关联。Django 提供了两个命令 **makemigrations** 与 **migrate**，当每次修改模型的定义时，必须执行这两条命令来进行数据库的迁移与同步。

即便现在没有定义任何模型类，但执行这两条命令后，Django 会自动生成一些系统默认表，这就代表着已经顺利连接上数据库啦。

### 5. 启动服务器

Django 为我们内置了一个轻量级的 web 服务器，便于开发者进行测试。我们可以通过以下命令来启动这个简易服务器：

```
python manage.py runserver[21]
```

### 5.1.3 项目目录说明

本项目根据已设计的功能模块，一共分为 4 个 App 应用，分别为 users、courses、teachers 和 operations。项目的目录结构图如图 5-1 所示。

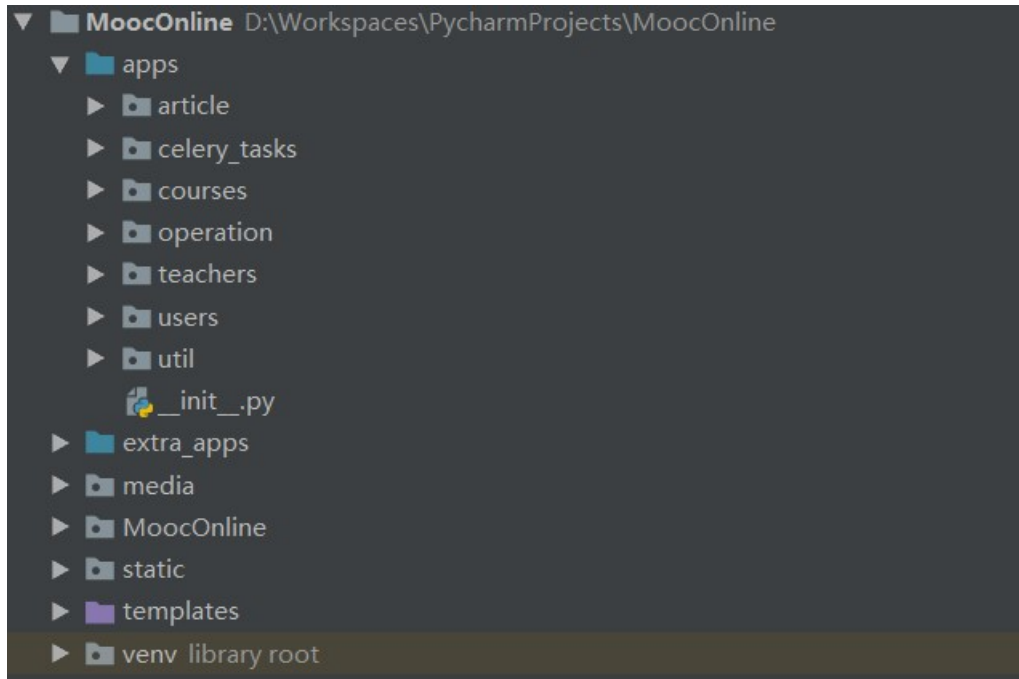


图5-1 项目的目录结构图

项目每个目录的作用如表 5-1 所示：

表 5-1 项目目录作用表

目录名	作用
apps	存放每个独立的 app 目录
extra_apps	存放外部引入的模块
users	用户功能模块的代码实现
util	邮件发送工具类实现
celery_tasks	定义 celery 异步任务
media	存放网站中的图片与视频等媒体资源文件
MoocOnline	存放项目的基本配置文件
static	存放项目 css、js 等静态文件
templates	存放前端 html 页面
venv	存放开发中使用的第三方库和包

## 5.2 学习者模块实现

### 5.2.1 注册登录功能实现

用户进入网站首页后，点击导航栏上的注册链接，进入注册页。注册页面见图 5-2。

The image shows a registration form titled "注册" (Register) in large, bold black text. Below the title are three input fields: "用户名" (Username), "邮箱" (Email), and "密码" (Password). To the right of the password field is a blue link that says "已经有账号了?" (Already have an account?). At the bottom center of the form is a blue button with the text "提交" (Submit).

图 5-2 注册页面

用户需要填写用户名、邮箱和密码，再点击提交完成注册。注册的表单填写规范如下：

用户名：不得为空，长度在 1-12 之间，由中文、英文、数字及下划线组成，且用户名不能重复

邮箱：不得为空，符合邮箱地址的格式，且不能输入已被注册的邮箱

密码：不得为空，长度在 6-12 之间

其中，检验邮箱是否注册的逻辑交给后端进行，当邮箱或用户名已经被注册了，系统会返回错误信息并显示“该邮箱已经被注册”，其它均可使用 jQuery

validate（jQuery 提供的表单验证插件）在前端校验完成。

注册成功后，系统会自动发送邮箱激活邮件，并跳转到邮箱激活邮件发送成功页。

由于发送邮件是一个极为耗时的操作，所以本平台使用 Celery 异步任务来完成邮件发送工作，以减少用户注册时等待的时间。

Celery 通过消息进行通信，通常使用一个叫 Broker 的中间人来协调 client（任务发出者）和 worker（任务处理者）。在本平台发送邮件的过程中，任务发出者、中间人与任务处理者都在同一台电脑上启动，当需要发送邮件时，注册程序将邮件发送任务发到任务队列中，任务处理者会对任务队列进行监听，当任务队列中新增了发送邮件的任务时，任务处理者会及时处理这个请求。

本文使用 Redis 作为 Broker（中间人），所以需要在 settings.py 文件中配置好中间人的基本信息。中间人的配置如下所示：

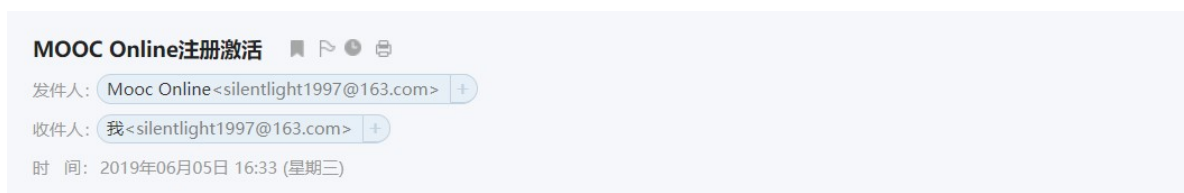
```
BROKER_URL = 'redis://127.0.0.1:6379/8'
```

```
BROKER_TRANSPORT = 'redis'
```

然后，需要在 send\_register\_active\_email() 方法前加上 celery 的注解“@app.task”，来定义一个任务函数。在调用发送邮件的方法时，需要在后面加上.delay()，比如 send\_register\_active\_email().delay()，这样发送邮件就会被当做一个异步任务执行了。最后，还需要启动任务处理者，如果任务处理者不启动，那么这个异步任务也就不会得到执行，启动任务处理者的命令如下所示：

```
celery -A apps.users.tasks worker --pool=solo -l info
```

这样配置完成后，就可以使用 celery 异步消息队列技术来发送邮件了。随后进入 163 邮箱收件箱，可以看到网站发来的激活邮件：



## silence, 欢迎您成为MOOC Online注册会员

请点击下面的链接激活您的账号: [http://127.0.0.1:8000/active/eyJhbGciOiJIUzUxMiIsImhhdCI6MTU1OTcyMzU3OSwiZXhwIjoxNTU5NzI4gk8YeWP47\\_m6ZRcelbFJB16iTREvcAGxw37pi4E-YsRZZdeVQbVVg](http://127.0.0.1:8000/active/eyJhbGciOiJIUzUxMiIsImhhdCI6MTU1OTcyMzU3OSwiZXhwIjoxNTU5NzI4gk8YeWP47_m6ZRcelbFJB16iTREvcAGxw37pi4E-YsRZZdeVQbVVg), 链接将在30分钟后失效

图 5-3 激活邮件发送成功页面

为了保障网站的安全性和稳定性,避免机器人或恶意用户在本平台注册激活大量的账号,需要对激活链接进行加密。本系统使用 Python 的第三方库 itsdangerous 加密与解密链接,itsdangerous 封装与实现了加密解密算法,且可设置加密的过期时间,简单易用。

用户点击邮件中提供的链接,系统后台对这个链接进行解密,如果能还原为加密前的数据,且未过到期时间,即可完成邮箱的激活。

激活成功后,用户就可以使用刚才注册好的账号进行登录。如果用户的账号还未经过邮箱的激活,则提示账号尚未激活;如果用户账号已经激活,但用户名与密码不匹配则登录失败;如果用户账号已经激活,且用户名和密码都正确,即可成功登入系统,进入首页。



图 5-4 系统首页

登录成功的流程可以总结如下：

- 1) 通过 POST 的方式提交登录表单，并接收用户名与密码两个参数
- 2) 调用 Django 提供的 `authenticate()` 方法进行登录校验
- 3) 校验通过，则调用 Django API 中的 `login()` 方法进行登录操作，将用户基本信息存入 session 会话中。
- 4) 返回首页

### 5.2.2 课程搜索功能实现

课程搜索的实现原理比较简单，即通过 GET 请求方式传递关键词参数，再进行模糊查询，当课程名、课程简介和课程描述中包含这个关键词时，则匹配成功。最后在课程列表页输出所有匹配的结果，课程搜索结果如图 5-5 所示。

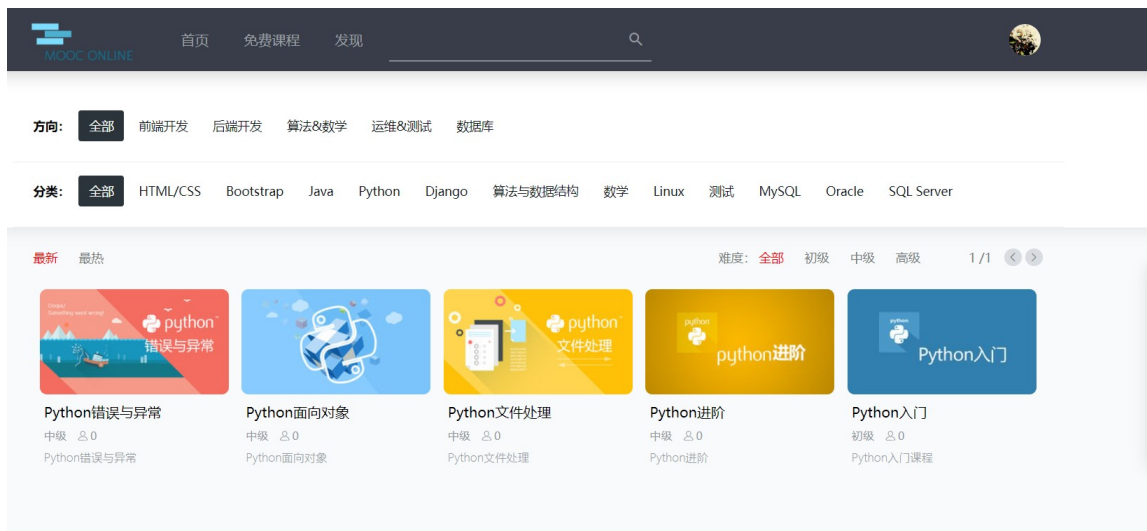


图 5-5 课程搜索结果页

课程搜索的关键实现代码如下所示：

```
keyword = request.GET.get('keyword', '')
if keyword:
    courses = courses.filter(
        Q(name__icontains=keyword)|
        Q(desc__icontains=keyword)|
        Q(detail__icontains=keyword))
```

### 5.2.3 课程学习功能实现

课程学习主要分为课程详情查看、课程视频观看、课程评论发表、课程笔记发布、课程 Wiki 学习、课程资料下载等功能。下面将分别对这些功能的实现展开介绍。

#### 1. 课程详情查看

课程详情页主要完成对包括课程基本信息、课程章节信息、课程教师信息、课程资料文件等信息的展示。课程详情页如图 5-6 所示。





图 5-6 课程详情页

当用户尚未学习课程的时候，课程详情页会显示蓝色的“开始学习”按钮，用户点击“开始学习”按钮后，开始学习课程的第一个视频；当用户已经学习本门课程的时候，课程详情页会显示蓝色的“继续学习”按钮，并在按钮的旁边显示用户本门课程的学习进度。课程开始学习与继续学习功能的关键代码如下：

```
if not user_courses:
    lesson = Lesson.objects.filter(course=course).first()
    if lesson:
        video = Video.objects.filter(lesson=lesson).first()
        user_courses = UserCourse(user=request.user,
                                   course=course, section=video)
        user_courses.save()
        course.stu_nums += 1
        course.save()
        video_id = video.id
        return HttpResponseRedirect('/course/video/%s/' % video_id)
```

else:

```
return HttpResponseRedirect('/course/info/%s/' % course_id)
```

else:

```
user_courses = UserCourse.objects.filter(user=request.user,
```

```
course=course).order_by("-add_time")[0:1].get()
```

```
video_id = user_courses.section.id
```

```
return HttpResponseRedirect('/course/video/%s/' % video_id)
```

## 2. 课程视频播放

课程视频页主要实现了课程视频的播放，具体页面如图 5-7 所示。课程视频播放实现的步骤是：

- 1) 在 URL 中传入视频 ID
- 2) 通过这个唯一的视频 ID，查询数据库中的视频表，并返回视频地址
- 3) 使用 HTML <video> 元素在 HTML 中嵌入视频播放器，并将视频地址填入播放器中，完成视频播放。

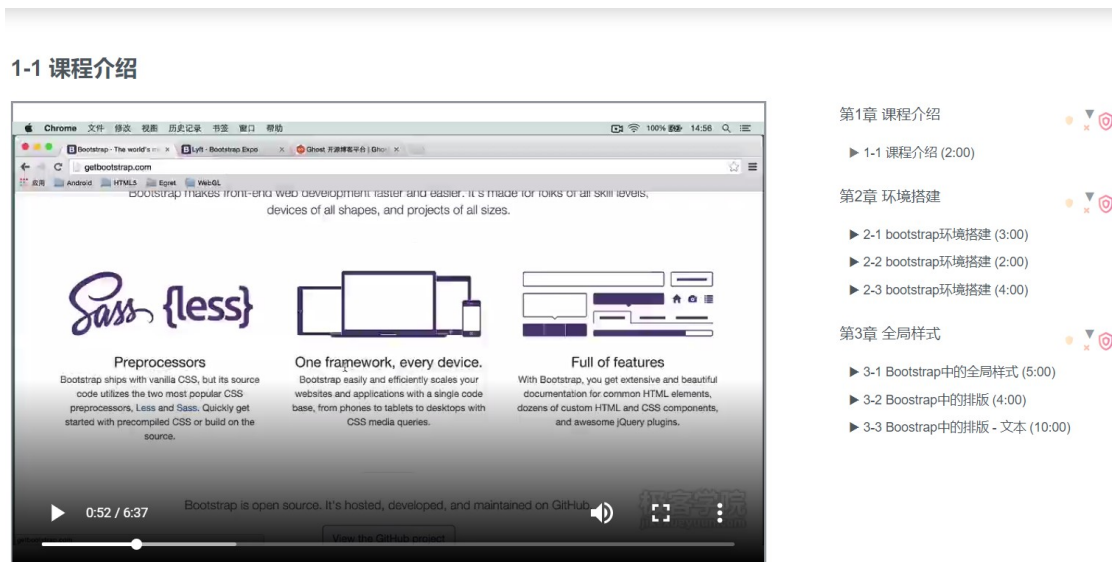


图 5-7 课程播放页

## 3. 课程 Wiki 学习

本部分主要实现课程的 Wiki 文档展示，Wiki 文档由 Markdown 语法编写，并渲染成 HTML 样式在前端页面显示，下面是渲染 markdown 文本的关键代码：

```
course_wiki.wiki = markdown.markdown
(course_wiki.wiki,extensions=[
    'markdown.extensions.extra',
    'markdown.extensions.codehilite',
])
```

渲染完成后，就可以在 HTML 页面上显示了：

```
{{ wiki.body|safe }}
```

Django 出于安全的考虑，会将输出的 HTML 代码进行转义，这使得 wiki.body 中渲染的 HTML 文本无法正常显示。管道符|是 Django 中过滤器的写法，而|safe 就类似给 wiki.body 贴了一个标签，表示这一段字符不需要进行转义了。

Wiki 文档页如图 5-8 所示：



图 5-8 课程 Wiki 页

#### 4. 课程资料下载

本部分需列出课程下的所有资料文件，并提供这些资料文件的下载链接。

Django 提倡将项目中上传的资源文件放到一个 media 文件夹下保存，需要进行的

配置如下：

```
MEDIA_URL = "/media/"
```

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media').replace("\\", "/")
```

这样，Django 在上传文件的时候，便会将文件放置到配置好的 media 路径下，获取资源的时候，只需要将这个根路径与数据库保存的文件相对路径拼接起来，就可以得到这个文件的下载链接了。

课程资料下载页如图 5-9 所示：



图 5-9 课程资料下载页

## 5. 课程评论发表

用户可以在文本框内输入不为空的文本内容，并点击发布问题按钮完成提交。

评论内容通过 Ajax 的方式提交，评论成功后如图 5-10 所示：

讨论区

手记

请写下你在学习本章节遇到的问题或疑惑

发布问题



silence

@admin 用的是3.3版本哦

时间: 2019年6月5日 19:13

1-1 课程介绍



admin

请问本课程使用的是哪个Bootstrap版本呢? Bootstrap 3.3还是4?

时间: 2019年6月5日 19:11

1-1 课程介绍

图 5-10 评论发布页

## 6. 课程手记发布

发布手记与发布评论的实现逻辑基本类似，也是通过 Ajax 方式提交表单，不同的是，发布手记可以使用富文本编辑器 **ckeditor** 来创建出格式丰富的笔记。手记发布成功后的页面如图 5-11 所示。

讨论区

手记

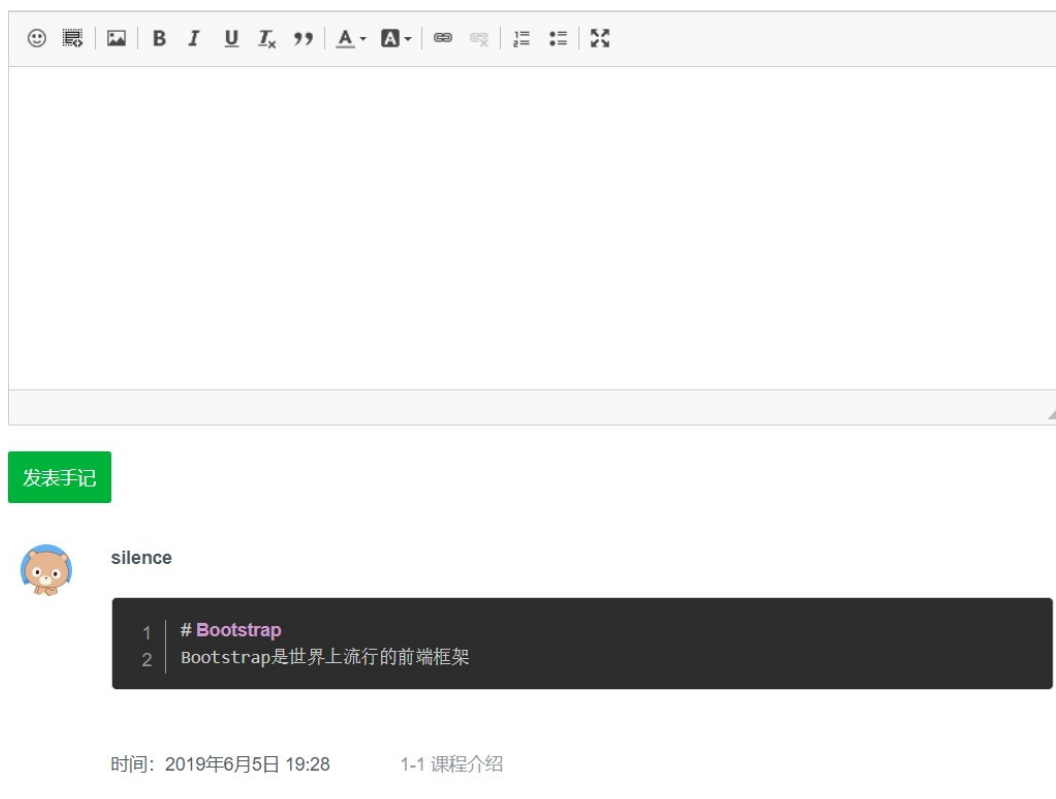


图 5-11 手记发布页

### 5.2.4 课程推荐功能实现

对于新注册的用户而言，因为该用户没有任何的学习记录，无法根据学习历史来推荐课程，所以，系统对新用户和老用户采用了不同的课程推荐算法。对新用户而言，推荐平台学习人数多的热门课程以及收藏人数多的优质课程；而对已经有学习记录的老用户，则需要根据其已学习过的课程来推荐一些他可能喜欢的课程。对于老用户，课程推荐算法是这样工作的：

- 1) 从用户课程表中筛选出课程 ID，即用户学习过的所有课程的 ID
- 2) 提取这些课程的所有分类，加入列表
- 3) 根据这些课程分类筛选出课程，并将这些数据填充成对象，交给前端展示

课程推荐结果如图 5-12 所示：

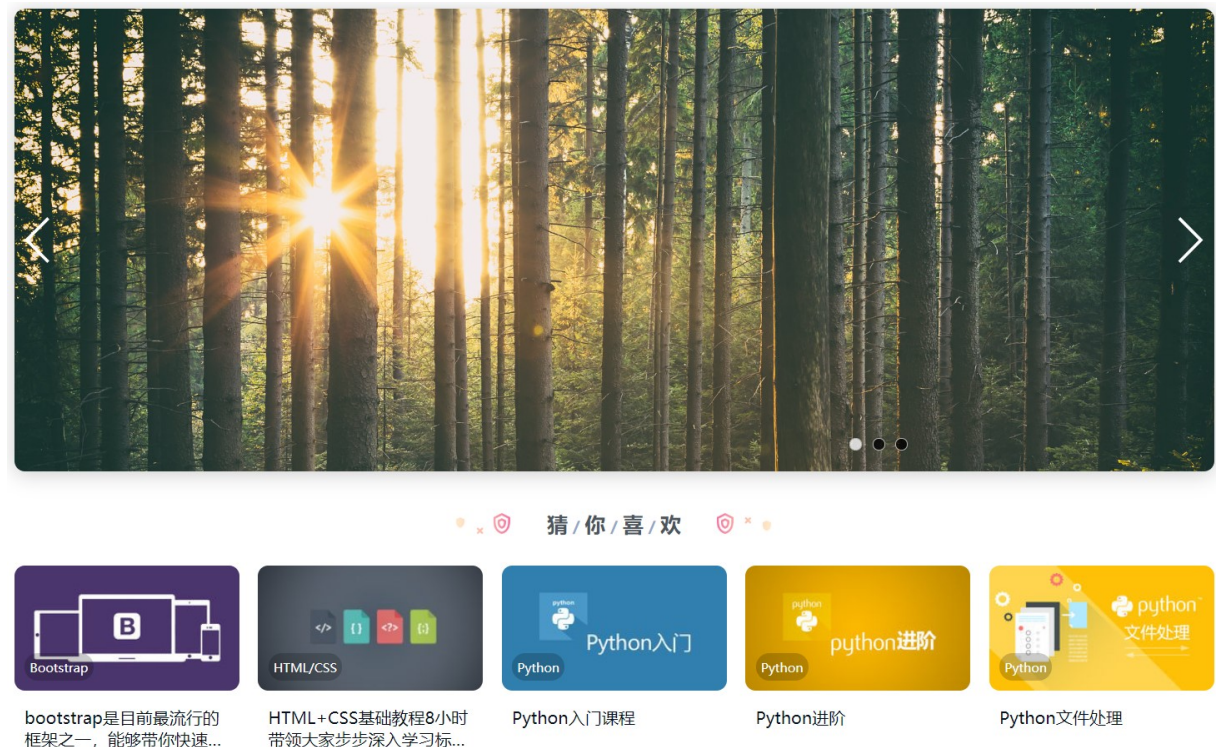


图 5-12 课程推荐结果

课程推荐算法的核心代码如下：

```
user_courses = UserCourse.objects.filter(user=request.user)
if user_courses:
    course_ids = [user_course.course_id for user_course in user_courses]
    allof_user_courses = Course.objects.filter(id__in=course_ids)
    course_tags = [course.classify_root_id for course in allof_user_courses]
    recommend_courses = Course.objects.filter(classify_root_id__in=
                                                course_tags)
    if recommend_courses.count()>5:
        recommend_courses = recommend_courses[:5]
else:
    recommend_courses = Course.objects.all().order_by(
        "-stu_nums").order_by("?")[:5]
```



### 5.2.5 个人中心功能实现

个人中心模块主要分为三个小功能，分别为个人资料查看及修改、我的课程和我的收藏。

个人资料页如图 5-13 所示：


首页 > 个人中心 > 个人信息

个人资料

我的课程

我的收藏

个人信息



修改头像

修改密码

昵 称:

职 位:

性 别: ☒ 男 ☐ 女

地 址:

手 机 号:

邮 箱:  [修改]

个性签名:

保存

图 5-13 个人资料页

在个人中心，学生用户可以直接对个人信息进行编辑，并点击保存按钮，即可修改个人信息；用户可以点击修改头像按钮，上传新的头像；用户还可以点击修改密码按钮，输入新的密码，完成密码的修改。

修改邮箱需要验证旧邮箱，系统会向用户的旧邮箱发送一个 4 位的随机字符验证码，用户需要输入正确的验证码才能完成邮箱的修改。当新的邮箱地址已注册时，显示错误信息。

用户还可以在个人中心查看学习过的课程，方便继续学习或再次学习课程。用户学习课程的记录保存在 operation\_usercourse（用户课程表）中，通过用户 ID 筛选即可查询出用户学过的所有课程，该部分的关键代码如下：



```
def get(self, request):
    u_courses = UserCourse.objects.filter(user=request.user)
    return render(request, "mycourse.html", {
        "u_courses": u_courses,
    })
```

首页 > 个人中心 > 我的课程



图 5-14 我的课程页

### 5.3 教师模块实现

课程上传模块，以及后面介绍的后台管理模块，都在后台管理系统进行，而本平台采用了 Xadmin 作为管理系统。Xadmin 是一个由国人定制开发的 Django 后台管理系统，该框架在 Django 自带管理系统的基础上添加了许多有用的插件，并把前端主题更换成界面美观简洁的 Bootstrap 主题。使用 Xadmin 后台管理系统，只需要简单的配置，便可对所有的数据表以可视化的方式完成 CURD（增删改查）操作，这样开发者不需要再开发出新的一套管理系统，提升了开发的效率。

教师登录后台管理系统后，即可看到自己教授的课程，如图 5-15 所示。为了让教师只能看到自己教授的课程，需要对课程列表进行筛选，筛选课程的关键代码如下：

```
def queryset(self):
```

```

qs = super(CourseAdmin, self).queryset()
if self.request.user.is_superuser:
    return qs
else:
    return qs.filter(teacher=self.request.user.teacher)

```



图 5-15 教师课程页

本平台使用 Xadmin 引入了较细粒度的权限控制，这体现在，教师只能查看课程的基本信息，而不能进行修改操作（课程管理由管理员完成）。

教师上传视频的操作如图 5-16 所示：

增加 视频

章节*	第3章 全局样式		
视频名*	3-4 Bootstrap的全局CSS样式		
视频地址*	<input type="button" value="选择文件"/> 【极客学院】一周...5162.mp4		
添加时间*	<input type="text" value="2019/06/10"/> <input type="button" value="今天"/>	<input type="text" value="4:39"/> <input type="button" value="现在"/>	
学习时长(分钟数)*	<input type="text" value="10"/>		

图 5-16 教师上传视频

## 5.4 后台管理模块实现

后台管理模块，分为用户管理、课程管理、教师管理、日志管理、轮播图管理和权限控制。其中用户管理、课程管理、教师管理、日志管理与上一小节中的阐述的管理操作类似，都是常见的增删改查操作，所以不再重复描述，下面将着重介绍权限控制功能。

因为教师是在学生用户的基础上添加了课程上传的权限，所以可以设置一个教师组权限，管理员在分配权限时就不必再一个个地添加单独的权限，减少了很多重复性的工作。管理员添加教师组权限的操作如图 5-17 所示，管理员对普通用户授予教师权限的操作如 5-18 所示。

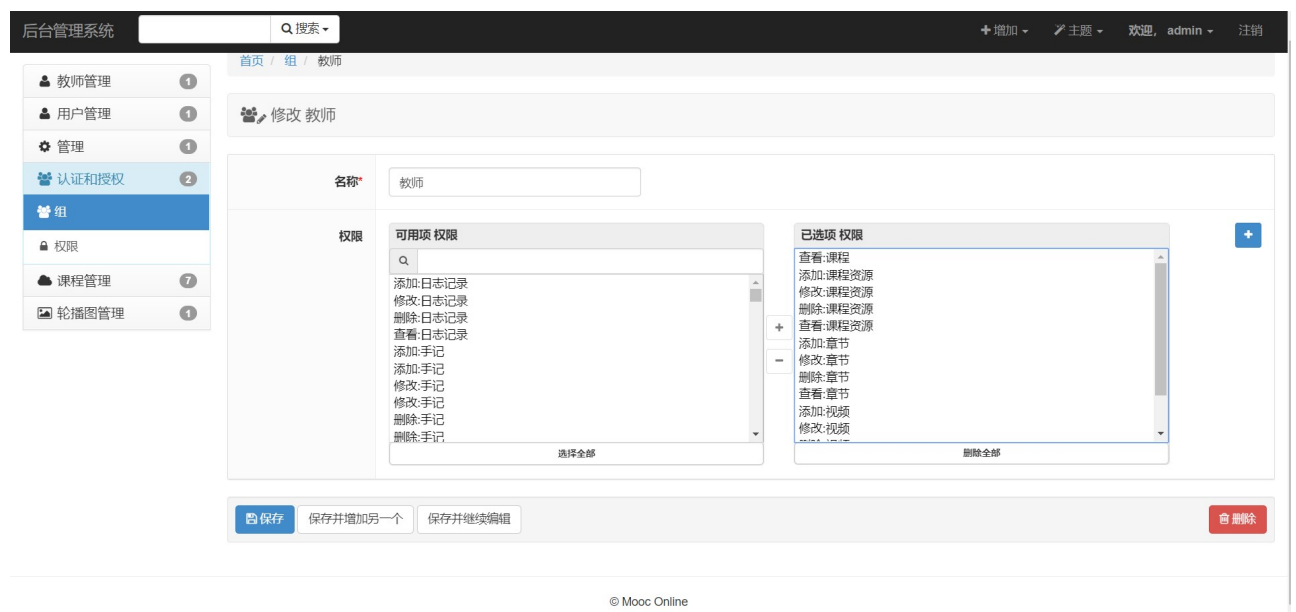


图 5-17 管理员添加教师组权限



图 5-18 管理员授予教师权限

## 5.5 本章小结

本章首先介绍了项目框架搭建的详细过程，再详细介绍了学习者模块、教师模块及后台管理模块的具体实现，并对一些重要的功能实现给出关键代码。

## 第六章 系统测试

软件测试是保证软件质量的重要手段，本章针对已开发完成的几个功能模块，设计良好的黑盒测试用例，以发现系统中出现的 bug，验证系统是否能满足客户的需求。

### 6.1 测试环境

对软件开发者而言，最简便的测试方法就是在本机上测试运行。以下是本系统的测试环境：

- (1) 测试机器：Windows 10（华硕 FL5800L）
- (2) 处理器：INTEL 酷睿 I7-5500U 2.40GHz
- (3) 内存（RAM）：4GB
- (4) 浏览器：Chrome，Firefox，Opera

### 6.2 功能测试

本节针对已开发完成的几个功能模块，设计有代表性的测试用例，来检测系统功能是否满足预期的效果，验证系统是否能满足用户的功能需求。下面将通过表格列出每个功能模块的测试用例及测试效果。

#### 6.2.1 注册登录模块测试用例

表 6-1 注册登录测试用例表

用例编号	测试操作	预期结果	测试结果
1	进入注册界面。用户名、邮箱、密码都置空，点击注册	提示不得为空	符合预期
2	用户名、邮箱、密码任一字段为空，点击注册	提示不得为空	符合预期
3	输入规范的用户名、密码、	注册完成，跳转到邮件发送	符合预期

用例编号	测试操作	预期结果	测试结果
	邮箱，点击注册	成功页	
4	输入已注册过的邮箱，点击注册	提示该邮箱已被注册的错误信息	符合预期
5	注册完成后，进入邮箱点击激活链接	跳转到登录界面并提示激活成功	符合预期
6	进入登录界面。用户名和密码都置空，点击登录	提示不得为空	符合预期
7	用户名和密码任意字段置空，点击登录	提示不得为空	符合预期
8	输入用户名、不正确的密码，点击登录	提示用户名与密码不匹配	符合预期
9	输入用户名和密码，但密码位数小于 6 位，点击登录	提示密码位数不得小于 6 位	符合预期
10	输入正确的用户名和密码，但账号尚未激活	提示账号尚未激活的错误信息	符合预期
11	输入正确的用户名和密码，且账号已经激活	登录成功，并跳转到网站首页	符合预期

## 6.2.2 课程搜索模块测试用例

表 6-2 课程搜索测试用例表

用例编号	测试操作	预期结果	测试结果
1	进入首页，定位到搜索栏，并直接回车	页面停留在首页，没有发生跳转动作	符合预期
2	在搜索栏中输入关键词，点击搜索图标	跳转到课程列表页，并显示全部符合关键词的课程结果	符合预期
3	在课程列表页点击分类链接	按照分类筛选课程，并正确显示结果	符合预期
4	在课程列表页按照时间、热度排序	能按给定的条件排序课程，并正确显示结果	符合预期

## 6.2.3 课程学习模块测试用例

表 6-3 课程学习测试用例表

用例编号	测试操作	预期结果	测试结果
1	进入课程详情页	能正确显示课程的基本信息	符合预期
2	进入课程详情页，点击我要学习	如果用户未登录，则跳转到登录页面；如果用户已登录，则跳转到课程第一个视频的播放页	符合预期
3	课程详情页，点击收藏图标	如果之前未收藏，收藏图标	符合预期

用例编号	测试操作	预期结果	测试结果
		变亮，如果已收藏，收藏图标变灰	
4	课程详情页，点击 Wiki	正常显示课程 Wiki，如果课程 Wiki 为空则显示空白	符合预期
5	课程详情页，点击课程资料下的链接	能成功下载对应的资料文件	符合预期
6	进入课程播放页	能正常加载视频，能调节音量，进度条，全屏播放	符合预期
7	课程播放页，点击视频下的讨论区	能正确显示该视频下的全部讨论	符合预期
8	课程播放页，添加一条空的讨论发言	提示不得为空	符合预期
9	课程播放页，添加一条不为空的讨论发言	添加成功并刷新页面，讨论可以正常显示	符合预期
10	课程播放页，点击视频下的手记	能正常加载一个富文本编辑器，显示该用户在该视频下的所有手记	符合预期
11	课程播放页，点击视频下的手记，添加一条空的手记	提示不得为空	符合预期
12	课程播放页，点击视频下的手记，添加一条不为空的手记	手记添加成功并刷新页面，手记可以正常显示	符合预期

## 6.2.4 课程推荐模块测试用例

表 6-4 课程推荐测试用例表

用例编号	测试操作	预期结果	测试结果
1	在首页的导航栏上，点击发现	如果用户未登录，则跳转到登录页面；如果用户已登录，进入发现页	符合预期
2	进入发现页	课程轮播图显示正常	符合预期
3	刚注册的用户，进入发现页	显示热门课程与优质课程	符合预期
4	已经有学习记录的用户，进入发现页	显示猜你喜欢与优质课程	符合预期

## 6.2.5 个人中心模块测试用例

表 6-5 个人中心测试用例表

用例编号	测试操作	预期结果	测试结果
1	在首页的导航栏上，点击头像	进入个人中心，正确显示用户个人信息	符合预期
2	进入个人中心，修改个人资料	验证新填写信息的有效性，	符合预期

用例编号	测试操作	预期结果	测试结果
	料	若有效则保存修改	
3	个人中心，点击我的课程	展示用户学习过的课程，若无则显示空白	符合预期
4	个人中心，点击我的收藏	展示用户收藏的课程，以及关注的教师	符合预期

## 6.2.6 课程上传模块测试用例

表 6-6 课程上传测试用例表

用例编号	测试操作	预期结果	测试结果
1	教师登录后台管理系统	能展示教师教授的课程	符合预期
2	教师对其教授的课程添加章节信息	添加章节成功	符合预期
3	教师管理其教授的课程章节信息	能进行增删改查操作，删除章节信息后章节下的课程也随之删除	符合预期
4	教师管理课程 wiki 信息	能进行增删改查操作	符合预期
5	教师上传视频	能成功上传视频文件，并提示添加成功	符合预期
6	教师管理视频信息	能进行增删改查操作	符合预期
7	教师上传课程资料文件	能成功上传课程资料，并提示添加成功	符合预期
8	教师管理课程资料信息	能进行增删改查操作	符合预期

## 6.2.7 后台管理模块测试用例

表 6-7 后台管理测试用例表

用例编号	测试操作	预期结果	测试结果
1	使用超级管理员账号登录后台管理系统，进行用户管理	能对用户信息进行增删改查操作	符合预期
2	进行课程管理	能对课程相关信息进行增删改查操作	符合预期
3	进行教师管理	能对教师相关信息进行增删改查操作	符合预期
4	权限控制，设置教师的组权限，并为教师用户分配权限	权限设置成功	符合预期
5	进行日志管理	查看、删除日志文件	符合预期



## 6.3 本章小结

本章主要介绍了系统的测试环境，并根据系统的功能模块设计并执行黑盒测试用例，测试每个功能模块的运行结果是否符合预期。

## 第七章 总结与展望

### 7.1 完成的工作

本文主要介绍了基于 Python 的在线教育平台的设计与实现。首先，研究在线教育平台的背景和意义，分析国内外研究现状；而后，调研在线教育平台使用的技术，如 Python、Django、MySQL、Bootstrap，设计系统的技术架构；其次，从用户的角度出发，并深入调研当前热门的在线教育平台，分析细化需求功能点；然后，通过之前的需求分析详细设计系统的每个功能模块，设计并生成数据库表；最后则是项目的编码与测试工作。

在项目开发过程中，本人在学习的过程中不断深入实践，熟练掌握了使用 Django 与 Bootstrap 技术开发网站的一般流程。最后，完成了所有功能需求，设计与实现出了一个满足预期目标的在线教育平台。

### 7.2 存在的问题及下一步工作

本人虽已完成了本平台所有功能的开发，但因为开发与精力有限，本平台还有许多不足之处需要改进：

1. 可以将 Django 框架更迭为 Django REST framework，实现前后端分离。Django REST framework 在 Django 框架的基础上添加了一些新特性，我们可以使用该框架设计出具有 RESTful 风格的 API，真正实现前后端的分离。
2. 网站前端界面美化。当前的前端界面主要基于 HTML、CSS、JavaScript、Bootstrap 等技术实现，未来可以使用 Vue、React 等流行的前端框架打造一个更加美观的前端界面。
3. 更精确的课程推荐算法。目前平台实现的课程推荐算法比较简单，可以通过引入基于用户过滤的推荐算法和基于物品过滤的推荐算法改进。

## 参考文献

- [1] 龙玲, 菅保霞. 国内典型在线教育平台学习支持服务现状分析[J]. 中国教育信息化, 2018 (12): 1.
- [2] 梁东, 黎军. MOOC 本地化对课程教学理念的研究实践[J]. 电脑知识与技术, 2018, 14(35): 138-139+154.
- [3] 郝丹. 国内 MOOC 研究现状的文献分析[J]. 中国远程教育, 2013, 11(21): 42-50.
- [4] 刘原铭. 基于 Python 的中小学云课堂平台设计与实现[D]. 北京交通大学, 2018.
- [5] McGuire R. The best MOOC provider: A review of Coursera, Udacity, and edX[J]. Ski. Com, 2014, 19.
- [6] 姜波. 基于 MOOC 模式学习平台的设计与实现[D]. 兰州大学, 2018.
- [7] 高婕梅, 韩骏, 刘菁, 等. MOOCs 平台比较与分析——以 edX, Coursera, Udacity 平台为例[D]. , 2014.
- [8] 张鸷远. “慕课”(MOOCs) 发展对我国高等教育的影响及其对策[D]. , 2014.
- [9] 张晶. 我国 MOOC 发展现状及展望[J]. 软件导刊, 2015, 14(1): 156-158.
- [10] 中商产业研究院. 2019-2024 年中国在线教育行业前景及投资机会研究报告 [DB/OL]. <http://www.askci.com/reports/20180312/1727055973830494.shtml>. 2019-04-24.
- [11] Lutz M. Learning python: Powerful object-oriented programming[M]. "O'Reilly Media, Inc.", 2013.
- [12] TIOBE. TIOBE Index for June 2019[DB/OL]. <https://www.tiobe.com/tiobe-index/>. 2019-06-06
- [13] 刘班. 基于 Django 快速开发 Web 应用 [J]. 电脑知识与技术, 2009, 5(07): 1616-1618.
- [14] 王鑫, 李磊. 慕课 MOOC 在线教育平台技术架构研究[D]. , 2014.

- [15] 向航. Django 架构纵览[DB/OL].  
<https://www.processon.com/view/572aca28e4b0c3c7496f2c0d>. 2016-05-05.
- [16] 吴卫平, 王丽芳, 蒋泽军. 基于 ORM 的数据持久层框架研究[J]. 微電子學與計算機, 2008, 25(7): 188-190.
- [17] Forta B. MySQL crash course[M]. Pearson Education India, 2005.
- [18] 夏亚丽. 基于 Django 的 O2O 家庭酒店商务平台的设计与实现[D].北京邮电大学,2017.
- [19] 大塚弘记. GitHub 入门与实践[M]. 第一版. 北京: 人民邮电出版社, 2015.
- [20] Bennett J. Practical Django Projects[M]. Apress, 2009.
- [21] Django Software Foundation. Writing your first Django app, part 1[DB/OL].  
<https://docs.djangoproject.com/en/2.2/intro/tutorial01/>. 2019-04-26

## 致谢

不知不觉已经在大学度过四个年头，完成这篇论文后就该对学校说再见了。如今再回首，虽然本人在大学期间留下了许多遗憾，但还是在工大学到了不少的知识。这短暂的四年间，浙江工业大学以其开放包容的胸襟见证了我的成长，对于工大“厚德健行”的校训，本人将会始终铭记并践行。

首先，我要衷心地感谢我的导师沈老师。沈老师无论从毕设题目的选择，还是系统开发上，再到最后毕业论文的编写，都给了我许多建设性的意见。在我懈怠与进度缓慢的时候，老师也会敦促我及时完成任务，确保了毕业设计能跟上进度。沈老师严谨的治学态度给我留下了深刻的印象，我会铭记老师对我的“老实做人，认真做事”的教诲，再次感谢沈老师无私的指导与帮助。

此外，我还要感谢在论文撰写过程中那些给予我帮助的人，有了他们，我的毕业论文才得以顺利完成。

最后，我还要感谢我的父母、家人以及朋友，是他们在经历人生低谷之时给予了我许多精神上的鼓励，让我能坚持把这篇毕业论文完成。

## 附录

附件 1 毕业设计文献综述

附件 2 毕业设计开题报告

附件 3 毕业设计外文翻译（中文译文与外文原文）