

学校代码：10904



**学 士 学 位 论 文**

基于Web的慕课管理系统的设计与实现

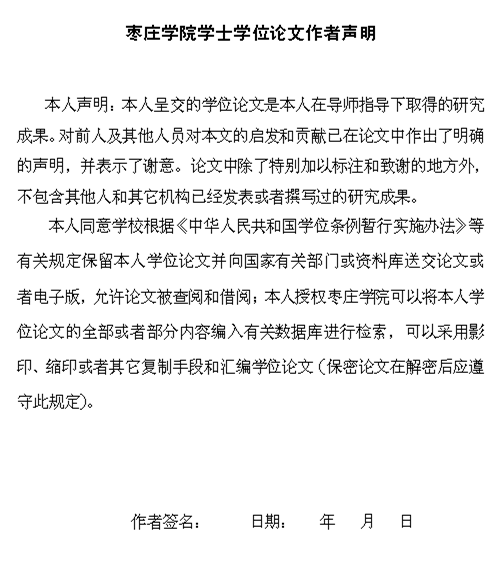
|  |  |
| --- | --- |
| **姓 名：** | 赵金龙 |
| **学 号：** | 201400110324 |
| **指导教师：** | 董西尚 副教授 |
| **学 院：** | 信息科学与工程学院 |
| **专 业：** | 计算机科学与技术 |
| **完成日期：** | 2018年4月30日 |



**学 士 学 位 论 文**

基于Web的慕课管理系统的设计与实现

|  |  |
| --- | --- |
| **姓 名：** | 赵金龙 |
| **学 号：** | 201400110343 |
| **指导教师：** | 董西尚 副教授 |
| **学 院：** | 信息科学与工程学院 |
| **专 业：** | 计算机科学与技术 |
| **完成日期：** | 2018年4月30日 |



摘 要

在“互联网＋教育”的背景下，将互联网和教育充分融合，便形成了慕课教育。在互联网时代，人们有足够的时间在网络当中，那么把教育应用到互联网，慕课教育便有了广阔的前景。传统的教育是定时定点定量的学习，而慕课教育却没有那么多严格的限定，只要有一台可以连接互联网的终端就可以进行学习。本系统立足于“互联网+”模式，设计并开发一套快捷高效，适用于高校和企业的慕课管理系统。结合现代互联网教育的实际情况，特决定建立慕课管理系统。

本系统后端采用Python作为开发语言，以Pycharm作为开发工具，设计一套相对完整的慕课管理系统。前端页面采用HTML+CSS+JavaScript，开发一套慕课管理系统，从而使各位讲师以及学校、企业更加便捷、高效的进行互联网教育。

慕课管理系统设计上采用B/S模式，建立基于Python语言、HTML语言MySQL数据库为核心的管理系统。根据基于Web的慕课管理系统的要求完成了应用系统的整体分析与设计。与传统的在线教育系统相比，本系统具有相对稳定便捷的使用方式。经过一系列的调试与进，基于Web的慕课管理系统是稳定的，并进行了真机测试和验证，基本满足客户的标准需求。

【关键词】慕课；Python；Pycharm；HTML；MySQL；Web

Abstract

In the context of "Internet + education", the Internet and education are fully integrated, so we have MOOCs. In the age of internet, many people live in the network, so MOOCs have broad prospects. The traditional education mode sets a finite limit to the amount and time. But the MOOCs has no restrictions. You can learn with a terminal that can connect to the Internet. The system based on the "Internet +". Design and develop a set of fast and efficient, suitable for college and company management system. Combined with the internet education situation, specifically decided to establish a MOOCs management system.

The background of the system using Python as the development language to Pycharm as a development tool, design a relatively complete menegement system. The front desk using HTML+CSS+JavaScript, the development of a Moocs management system, so that lecturers,college and company, accurate and efficient using Internet education.

MOOCs management system design using B/S mode,Python language,HTML language,MySQL database as the core of the manegement system. According to the requirements of the application of the MOOCs management system based on the browser platform, the overall analysis and design of the student attendance system application is completed. Compared with the traditional way of the Internet education the system has a relatively stable and convenient way of Internet education. After a series of debugging and progress, based on Web platform MOOCs management system is stable, and carried out a real machine test and verification, basically meet the user's standard needs.

【key words】MOOCs; Python;Pycharm; HTML; MySQL; Web

目 录

[第1章 绪 论 1](#_Toc513809761)

[1.1 研究背景 1](#_Toc513809762)

[1.2 国内外研究现状 1](#_Toc513809763)

[1.3 背景意义 2](#_Toc513809764)

[1.4 系统特征 2](#_Toc513809765)

[1.5 论文组织结构 2](#_Toc513809766)

[第2章 系统相关技术介绍 4](#_Toc513809767)

[2.1 Python语言简介 4](#_Toc513809768)

[2.2 Django框架简介 4](#_Toc513809769)

[2.3 MySQL数据库简介 5](#_Toc513809770)

[2.4 B/S体系架构 5](#_Toc513809771)

[第3章 系统分析 6](#_Toc513809772)

[3.1 可行性分析 6](#_Toc513809773)

[3.1.1 技术可行性 6](#_Toc513809774)

[3.1.2 经济可行性 6](#_Toc513809775)

[3.1.3 操作可行性 6](#_Toc513809776)

[3.2 需求分析 6](#_Toc513809777)

[3.2.1 系统功能模块分析 6](#_Toc513809778)

[3.2.2 数据需求分析 9](#_Toc513809779)

[第4章 总体设计 10](#_Toc513809780)

[4.1 系统功能总体设计 10](#_Toc513809781)

[4.1.1 用户功能 10](#_Toc513809782)

[4.1.2 管理员功能图 10](#_Toc513809783)

[4.1.3 整体功能需求 11](#_Toc513809784)

[4.2 业务流分析 11](#_Toc513809785)

[4.3 数据库设计 13](#_Toc513809786)

[4.3.1 数据库字典 13](#_Toc513809787)

[4.3.2 数据库概念结构设计 15](#_Toc513809788)

[4.3.3 数据库逻辑设计 17](#_Toc513809789)

[第5章 系统详细设计及实现 21](#_Toc513809790)

[5.1 网页前端设计 21](#_Toc513809791)

[5.1.1 登录功能 21](#_Toc513809792)

[5.1.2 注册功能 22](#_Toc513809793)

[5.1.3 个人中心功能 23](#_Toc513809794)

[5.1.4 课程学习功能 24](#_Toc513809795)

[5.1.5 课程评论功能 25](#_Toc513809796)

[5.1.6 收藏功能 28](#_Toc513809797)

[5.2 后台管理设计 32](#_Toc513809798)

[5.2.1 后台主要功能 32](#_Toc513809799)

[5.2.2 信息添加功能 32](#_Toc513809800)

[5.2.3 信息维护功能 33](#_Toc513809801)

[第6章 系统测试 35](#_Toc513809802)

[6.1 系统测试目的 35](#_Toc513809803)

[6.2 主要功能模块测试 35](#_Toc513809804)

[6.2.1 用户登录功能测试 35](#_Toc513809805)

[6.2.2 用户学习功能测试 36](#_Toc513809806)

[6.2.3 用户收藏功能测试 37](#_Toc513809807)

[6.3 测试总结 38](#_Toc513809808)

[第7章 系统总结与展望 39](#_Toc513809809)

[参考文献 40](#_Toc513809810)

[致 谢 42](#_Toc513809811)

第1章 绪 论

1.1 研究背景

教育资源分布不均衡，影响了教育公平，成为制约当前教育发展的主要问题之一[1]。慕课教育却轻松解决了教育不公平的问题。2012年被称为全球慕课行业发展元年，但是慕课教育在国内起步稍晚。在2014年，国内拥有150万的慕课用户；但在随后的几年时间里，国内慕课行业迅速发展，至2015年底，国内拥有575万人的慕课用户，增长速度高达283%；截至2016年10月，国内已经拥有超过1000万的慕课用户。

国内慕课行业发展受到教育部先后出台的多项鼓励性政策推动。2015年，教育部出台《教育部关于加强高等学校在线开放课程建设应用于与管理的意见》推动及促进高等院校参与慕课课程制作、服务平台建设，并提倡相关服务创新。《教育部2016年工作重点》也提到充分利用市场机制建设推进包括慕课在内的在线开放课程的发展及应用。在2016年10月，《2016中国慕课行业白皮书》由国家教育部在线教育研究中心发布。该白皮书指出，国内慕课用户主体集中在主要分布在一二线城市，用户群体为18到35岁中青年群体。其中，18岁至25岁占47.9%，26岁至30岁占20.4%，31岁至35岁占15.2%。从未来的慕课平台发展来看，慕课教育将使学校、企业和用户能发挥更大的价值，并产生深远的影响。可以预见的是，随着云技术的发展，由于慕课以较低的学习成本为全球学习者提供了较优质的学习资源，从而在较大程度上改变了人们的学习方式以及教学方式，在促进全球高校交流、竞争的同时也实现了优质教育资源的建设与共享，必将导致慕课的影响力不断加强，其行业规模也将不断扩大。

1.2 国内外研究现状

在慕课教育蓬勃发展之前，网络课程资源建设已经在国内高等教育领域开展过多次。例如在2000年面向高校的“新世纪网络课程建设工程”，2003年启动的“国家精品课程建设工程”。截至到2010年，教育部已经评选出众多的国家精品课程、省级精品课程和校级精品课程。从2010年开始，网上开始流行Yale等知名高校的视频公开课，国内众多媒体比如网易，为这些视频公开课开辟了专门的频道，即现在的网易云课堂，该网站采用Java语言编写。除此之外教育部在2011年启动了“国家精品开放课程建设工程”，在精品课程的基础上进一步开放共享。

在慕课出现之前，在线教育在国内已经有一定的基础。在2012年慕课飞速发展之年，国内诸多高校纷纷跟进。清华大学、北京大学、复旦大学、上海交通大学等大学分别加入Courser和edX等慕课平台。2013年，清华大学在open-edX开源代码的基础上，构建了自主的慕课平台“学堂在线”。2014年“好大学在线”由上海交通大学推出，并且为自主研发。

1.3 背景意义

随着慕课教育飞速发展，帮助讲师适应变化，为他们提供在线教育平台，成为很多学校和机构的共同努力方向[2]。慕课为我们带来了很多的可能性，可能会使中国的教育发生变化，有望解决中国教育多年来的问题：推动现有高等教育质量的提升，解决很大一部分人上不了大学的问题，解决高等教育资源不均衡问题[3]。对于学校和机构而言，慕课系统可以方便对课程进行在线教育，更方便对网络课程的管理；对于学生而言，慕课系统使学生利用空闲时间进行学习，激发学生的学习兴趣。慕课系统更好的将网络技术应用在学生的学习生活中，切实为学生的学习带来便捷。同时，也为学校和企业对进行慕课教学提供了技术支持，提升学校的教学质量和企业的工作效率。

1.4 系统特征

系统将使用Pycharm作为本项目开发以及测试的环境，选用安全可靠的[MySQL](http://baike.baidu.com/view/24816.htm)为本项目的数据库管理工具，采用Navicat对数据库进行图形化管理。本项目业务逻辑使用Pycharm开发，采用B/S架构以增强系统的适应性。本项目前端页面采用HTML开发，HTML是最常用的Web页面编程语言，适用于系统前端页面的搭建。业务逻辑采用Python的Web框架Django进行开发。

硬件环境：PC机。

软件环境：Pycharm、Navicat、MySQL、Python、Django。

1.5 论文组织结构

本论文共分七章，具体安排如下：

1. 第1章为绪论，这一章主要讲述了本系统的研究背景与目标、研究内容、前景意义和论文的组织结构。

2. 第2章主要介绍了开发本系统的相关技术基础，包括Python、Django、MySQL的基础知识以及说明了系统说明、开发计划和B/S体系结构。

3. 第3章是对本系统做出系统分析，主要从可行性及系统的需求两个方面进行的分析。可行性分析从技术、经济、操作三个方面分析确定了系统开发的作用和意义。需求分析则确定了系统具体应该实现的功能。

4. 第4章介绍开发此系统的概要设计以及数据库的概要设计，系统的概要设计包括业务流分析、数据流分析、数据字典，数据库的设计包括数据库的逻辑结构设计和物理结构设计。

5. 第5章介绍系统的详细设计，包括设计目标、逻辑结构设计、具体实现的功能等。

6. 第6章介绍系统测试。本文主要测试登录功能、学习功能、收藏功能。

7. 最后一章是对本系统的总结与对未来发展和改善的展望。

第2章 系统相关技术介绍

2.1 Python语言简介

Python 语言是一种免费开源的编程语言。Python是一种解释型的、通用的高级编程语言。设计理念强调代码可读性[4]。由龟叔（荷兰人）在1989年为了打发无聊时间而发明，第一个公开发行版发行于1991年。Python是开源的，遵循 GPL(GNU General Public License)协议。其语法简单方便，使用空白符作为语句缩进是其一大特点。Python具有丰富和强大的库。它常被昵称为胶水语言，能够把用其他语言制作的各种模块（尤其是C/C++）很轻松地联结在一起[5]。

由于Python语言的简洁性、易读性以及可扩展性，在国外用Python做科学计算的研究机构日益增多，一些知名大学已经采用Python来教授程序设计课程。Python语言简单的特点，及其大量的开源项目、众多的扩展库形成了Python广阔的生态环境。对工程技术，科研人员处理实验数据、制作图表，甚至开发科学计算应用程序都具有良好的适应性。基于Web的 Python编程环境, 更多地依赖于Python的方便性, 并且Python在入门框架选择时, 更加简便[6]。

2.2 Django框架简介

Django 是一个常见的 Python Web 应用框架。它是开源的Web开发框架，包含了多种组件，可以实现关系映射、动态内存管理、界面管理等功能[7]。它是一个鼓励快速开发和干净，实用的框架设计。采用了MTV的框架模式，即模型M，模板T和视图V。在2005年7月遵循BSD协议进行开源。Django可以更容易地快速构建更好的Web应用程序，并且它使用更少的代码。

Django框架可以运行在多种网页服务器上，比如Apache、Nginx。并且支持Oracle,Postgresql,MySql,Sqlite3等数据库。Google App Engine也支持Django的某些部分，国内支持的平台有（SAE）Sina App Engine，（BAE）百度应用引擎。

Django自带了以下设计原则：

1、松耦合−Django的目的是使其堆栈中的每个元素独立于其他。

2、较少编码−较少代码，所以能迅速的开发。

3、不必再重复(DRY)−一切都应只在一个地方，而不是一次又一次地重复来开发它。

4、更快的开发−Django的理念是尽一切所能，便于超高速开发。

5、干净的设计−Django严格维护一个干净的设计在其自己的代码，并可以很容易地遵循最佳Web开发实践

2.3 MySQL数据库简介

MySQL 是一种关系型数据库管理系统。因为MySQL简单易用，所以很多企业都在使用它来构建自己的数据库。MySQL具备以下这些优点：

基于开源许可发布，可免费使用；自身的功能非常强大；使用标准的SQL数据库语言；可以跨平台运行，支持多种语言，包括 PHP、PERL、Python、C、C++ 及 Java 等语言对大型数据集支持性好。由于开源的特性，可以对MySQL进行自定义的修改，以便获得更高的性能和适应更广泛的开发环境。通过 Python 访问 MySQL 数据库能充分发挥 Python 和 MySQL 数据库的优势[8]。

2.4 B/S体系架构

B/S 体系结构，是指浏览器/服务器结构。是互联网时代发展的主流架构。B/S架构是对C/S架构的改进，为区别于C/S所以成为B/S架构。主要是利用不断成熟的浏览器技术，结合浏览器的多种脚本语言，很少部分的事务逻辑在B端运行，主要的事务逻辑在S端运行，形成了三层结构。用浏览器就实现了原来需要复杂的专用软件才能实现的强大功能，并节约了开发成本，是一种全新的软件体系结构[9]。

B/S具有是维护简单、开发迅速、适应性和共享性强、总体拥有成本低。虽然B/S有众多的优点，但是也存在一定的缺陷。比如数据安全性问题、对服务器要求过高、数据传输速度慢、软件的个性化问题；对某些特殊功能难以实现；很难构建大型复杂的应用。

第3章 系统分析

3.1 可行性分析

3.1.1 技术可行性

慕课管理系统的开发基于B/S模式，主要包括前端页面的开发以及后端业务逻辑和管理系统的建立和维护两个方面。本系统采用Pycharm作为后端的开发工具，利用开源的Python Web框架——Django进行开发后台业务逻辑[10]；前端页面使用开源HTML代码。业务逻辑处理获得数据并把数据存入数据库。

3.1.2 经济可行性

对开发这个项目的经济要求比较低。同时，软件开发所需要的软件、硬件系统也方便获得。使用Python开发后台，使用HTML开发前端，成本不高，原则上可以实现。系统的维护是整个系统中最重要的，这个项目是可以进行实时维护，节约人力物力财力。所以系统开发具有经济可行性，也是有效的。

3.1.3 操作可行性

由于大量使用开源项目，使得项目的开发迅速便捷。普通用户只要有一台能够连接互联网的终端设备就能够进行使用和学习。用户界面清晰，操作方便，非常适用于在线学习。本系统具有完整的管理功能，可以对网站各项内容进行操作、管理和维护，操作十分方便。

基于上述三个方面，该系统具有高开发可行性，在技术上和经济上的操作。因此，该系统可以进行设计和开发工作。

3.2 需求分析

3.2.1 系统功能模块分析

本系统根据在线学习实际情况来设计页面，操作简单容易，便于学生进行学习。本系统前端系统主要有用户注册、用户登录、用户收藏、用户学习课程、用户课程评论、用户个人中心等功能[11]。本系统后端管理系统主要有用户管理、课程管理、讲师管理、机构管理、网站首页相关管理[12]。

1. 用户功能如下：

(1) 用户注册：

用户需要注册才能使用本系统，通过填写的自己的邮箱、密码进行注册，只有注册之后才能进行登录。

(2) 用户登录：

用户在进行注册之后必须要进入注册邮箱进行激活，激活后使用自己注册的邮箱和密码进行登录，之后才能进行一些其他功能的使用。

(3) 用户收藏：

用户登录后，通过网页浏览本网站数据库当中的内容。并可以对自己喜欢的存储的课程、讲师、机构进行收藏。

(4) 用户学习课程：

用户登录后在浏览课程时可以通过点击学习按钮对课程进行学习。

(5) 用户课程评论：

用户在进入某课程进行学习时，可以对课程进行评论。

(6) 用户个人中心：

注册用户可以通过该功能对自己的个人基本信息，进行增删改查。

2. 管理员功能如下：

(1) 用户管理：

该功能实现对用户的信息管理和权限管理。

(2) 课程管理：

该功能实现对系统内所有的课程进行管理，包括课程本身内容的管理和课程属性的管理。

(3) 讲师管理：

该功能实现对系统内所有的讲师进行管理，包括讲师本身信息管理和讲师属性的管理。

(4) 机构管理：

该功能可以对所有的机构进行管理。

(5) 网站首页相关管理：

该功能可以对网站首页的相关元素进行管理。

通过对系统功能的分析，完成用户功能用例图，如图3-1所示。



图3-1 用户功能用例图

通过对系统功能的分析，完成管理员功能用例图，如图3-2所示。



图3-2 管理员功能用例图

3.2.2 数据需求分析

在性能需求分析方面，本系统在用户的使用和管理员管理上有着十分清晰的界限，管理员与用户有着不同的页面。从功能需求分析中得到，要实现主要是用户管理、用户学习课程、用户课程评论、课程管理、讲师管理、机构管理[13]。对这主要功能进行数据的需求分析：

1. 用户管理：用户注册时将信息存入用户表，用户登录时验证用户信息以及用户个人基本信息的存储和显示。

2. 用户学习课程：用户在课程页点击学习后便与课程形成了联系存入数据表中。

3. 用户课程评论：存储并显示学习过该课的学生对该课程的评价。

4. 课程管理：存储并显示课程的相关内容和属性。

5. 讲师管理：存储并显示讲师的信息。

6. 机构管理：存储并显示机构的相关内容和属性。

第4章 总体设计

4.1 系统功能总体设计

4.1.1 用户功能

对于一款基于Web的慕课管理系统，最核心的功能就是实现课程的在线学习。为实现在线学习功能，本系统应支持用户管理、课程管理等基础功能。为保证用户信息的一致性、可操作性和安全性，注册界面中用户需要填写自己的邮箱、密码和验证码，点击注册按钮后系统将用户信息提交到服务器，服务器接收到用户注册信息并发送验证邮件，用户需要点击验证邮件，验证通过后便可以进行登录[14]。登录界面中，用户填写自己的邮箱和密码即可点击登录按钮，系统将验证用户是否存在以及用户信息的标志位。用户功能图如图4-1所示。



图4-1 用户功能图

4.1.2 管理员功能图

管理员具有的功能有用户管理、课程管理、机构管理、讲师管理等功能。还应该具有网站首页相关管理等功能。管理员功能如图4-2所示。



图4-2 管理员功能图

4.1.3 整体功能需求

根据前面章节需求分析的情况，结合本系统所要实现的功能，具体设计方案如图4-3所示。

图4-3 整体功能图

4.2 业务流分析

慕课管理系统工作流程为：打开网站进入慕课系统首页，界面右上角显示用户可以登录或者注册，若点击立即注册，界面跳转到注册界面。在注册界面输入邮箱、密码、验证码即可实现注册，注册后跳转到登录界面。用户在登录界面填写邮箱和密码进行登录，数据验证正确后，网页跳转到网站首页界面。此时用户可以浏览网站中存储的相关信息。在相关信息页面可以进行相关操作。

管理员可以通过后台管理系统对网站内部信息进行管理。管理用户信息；管理课程信息，对课程的添加、课程信息查看和修改、课程评论的查看和修改、章节信息的查看和修改；管理机构信息，对机构信息查看和修改；管理讲师信息，对讲师信息查看和修改[15]。本系统的业务流程图如图4-4所示。

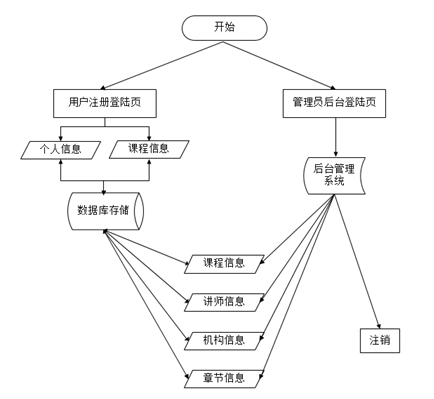


图 4-4 系统业务流程图

本系统初步设计一个准确方便快捷的慕课管理系统，以利于用户进行相关的学习等操作。为了使系统在用户使用中发挥更大的作用，实现在线教育的程的方便化，智能化，提高开发效率和工作质量，现提出如下系统开发目标：

1. 操作简单：完全模块式的界面布局，使得用户的信息浏览与用户的学习更加简单方便。

2. 实时可见：对用户学习的课程，用户可以在个人中心清晰的显示出来。

3. 功能完善：不仅实现基本的用户注册、用户登录、课程管理等功能，完成系统对用户学习的基本要求。而且，系统还能对课程评论、下载课程附件，讲师管理，机构管理，给用户良好的体验和感受。

4. 动态管理：对系统数据库信息实行动态操作，能实现对数据库内容的实时查询、实时更新，实时修改和动态录入数据。

4.3 数据库设计

4.3.1 数据库字典

1.数据字典列表——数据流

每条数据流记录由数据存储名称、数据来源、数据去向、数据组成、描述五部分组成，以此来清楚的了解每条数据流的具体信息。如表4-1所示。

表 4-1 数据流表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据名称 | 数据来源 | 数据去向 | 数据组成 | 描述 |
| 用户信息 | 注册界面 | 用户表 | 用户信息=用户ID（主键）+昵称+生日+地址+手机号码+头像+邮箱+密码+状态 | 存储了用户信息 |
| 课程信息 | 课程添加页面 | 课程展示界面 | 课程信息=课程ID（主键）+课程名称+课程描述+课程详情+课程等级+学习时间+学习人数+收藏人数+封面图+点击数+课程类别+课程标签+课程内容+添加时间+机构（外键）+讲师（外键） | 存储了课程信息 |
| 机构信息 | 机构添加页面 | 机构展示界面 | 机构信息=机构ID（主键）+机构名称+机构描述+机构标签+机构类别+点击数+收藏数+封面图+机构地址+所在城市+学习人数+课程数+添加时间 | 存储机构信息 |
| 讲师信息 | 讲师添加界面 | 讲师展示页面 | 讲师信息=讲师ID（主键）+讲师名+工作年限+就职公司+公司职位+教学特点+年龄+点击数+收藏数+添加时间+头像+所属机构（外键） | 存储讲师信息 |
| 章节信息 | 章节添加页面 | 课程详情页面 | 章节信息=章节ID（主键）+章节名+添加时间+所属课程（外键） | 存储章节信息 |
| 视频信息 | 视频添加页面 | 章节列表页面 | 视频信息=视频ID（主键）+视频名称+视频长度+访问地址+添加时间+所属章节（外键） | 存储视频信息 |

2.数据字典列表——数据处理

每条数据处理记录由名字、描述、输入、过程、输出五部分组成，此表记录了各个模块数据处理事件的详细信息。见表4-2所示。

表4-2 数据处理

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名字 | 描述 | 输入 | 过程 | 输出 |
| 用户登录 | 用户在登录界面登录 | 邮箱、密码 | 判断是否为合法的用户 | 网站首页 |
| 课程信息 | 课程的详细信息 | 输入要添加的课程信息 | 系统获取网页提交的课程信息是否合法，如果合法发送到数据库 | 课程信息保存是否成功 |
| 讲师信息 | 讲师的详细信息 | 输入要添加的讲师信息 | 系统获取网页提交的讲师信息是否合法，如果合法发送到数据库 | 讲师信息保存是否成功 |
| 机构信息 | 机构的详细信息 | 输入要添加的机构信息 | 系统获取网页提交的机构信息是否合法，如果合法发送到数据库 | 机构信息保存是否成功 |
| 章节信息 | 详细的章节信息 | 输入要添加的章节信息 | 系统获取网页提交的章节信息是否合法，如果合法发送到数据库 | 章节信息保存是否成功 |
| 视频信息 | 详细的视频信息 | 输入要添加的视频信息 | 系统获取网页提交的视频信息是否合法，如果合法发送到数据库 | 视频信息保存是否成功 |

3.数据字典列表——数据存储

每条数据存储记录由名字、描述、数据组成、位置、输出五个部分组成，此表记录了数据存储的详细信息。见表4-3所示。

表4-3 数据存储

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名字 | 描述 | 数据组成 | 位置 | 输出 |
| 数据库信息 | 存储数据库的基本信息 | 数据库信息=用户信息+课程信息+机构信息+讲师信息 | 数据库 | 查询得到的信息界面 |

4.3.2 数据库概念结构设计

根据前面的设计规划出的实体有：用户实体、课程实体、机构实体、讲师实体。各个实体的具体描述E-R图如下。



图4-5 用户实体图



图4-6 课程实体图



图4-7 机构实体图



图4-8 讲师实体图



图4-9 章节实体图



图4-10 视频实体图



图4-11 全局E-R图

4.3.3 数据库逻辑设计

数据库逻辑结构设计需要将上面的数据库概念结构转化为MySQL数据库系统所支持的实际数据模型，也就是数据库的逻辑结构。在实体以及实体之间关系的基础上，形成数据库的表格以及各个表格之间的关系。主要数据表如下：

usersprofile表主要存储系统用户的个人信息，包括邮箱，密码，昵称等等，具体的表结构设计如表4-4所示。

表4-4 user表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 大小 | 允许为空 | 是否主键 | 备注 |
| 1 | id | int | 11 | 否 | 是 | 用户ID |
| 2 | password | varchar | 128 | 否 | 否 | 密码 |
| 3  5 | nick\_name | varchar | 50 | 是 | 否 | 昵称 |
| 4 | birday | date |  | 否 | 否 | 生日 |
| 5 | address | varchar | 100 | 否 | 否 | 地址 |
| 6 | gender | varchar | 6 | 否 | 否 | 性别 |
| 7 | mobile | varchar | 11 | 是 | 否 | 手机号码 |
| 8 | email | varchar | 254 | 否 | 否 | 邮箱 |
| 9 | image | varchar | 100 | 否 | 否 | 头像 |
| 10 | is\_active | tinynit | 1 | 否 | 否 | 状态 |

course表主要存储课程信息，有课程名称，课程描述，课程详情等基本信息，管理员可以在后台管理系统将信息直接写入表中，查看时直接从表中调取数据。具体的表结构数据如表4-5所示。

表4-5 course表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 大小 | 允许为空 | 是否主键 | 备注 |
| 1 | id | int | 11 | 否 | 是 | 课程ID |
| 2 | name | varchar | 50 | 否 | 否 | 课程名称 |
| 3 | desc | varchar | 300 | 否 | 否 | 课程描述 |
| 4  5 | detail | longtext |  | 否 | 否 | 课程详情 |
| 5 | degree | varchar | 2 | 否 | 否 | 课程等级 |
| 6 | learn\_times | int | 11 | 否 | 否 | 学习时长 |
| 7 | students | int | 11 | 否 | 否 | 学习人数 |
| 8 | fav\_nums | int | 11 | 否 | 否 | 收藏人数 |
| 9 | image | varchar | 100 | 否 | 否 | 封面图 |
| 10 | click\_nums | int | 11 | 否 | 否 | 点击数 |
| 11 | category | varchar | 20 | 否 | 否 | 课程类别 |
| 12 | tag | varchar | 10 | 否 | 否 | 课程标签 |
| 13 | youneed\_know | varchar | 300 | 否 | 否 | 课程须知 |
| 14 | teacher\_tell | varchar | 300 | 否 | 否 | 课程内容 |
| 15 | add\_time | datetime | 6 | 否 | 否 | 添加时间 |
| 16 | teacher\_id | int | 11 | 是 | 否 | 讲师外键 |
| 17 | course\_org\_id | int | 11 | 是 | 否 | 机构外键 |

courseorg表中主要存储课程信息，有机构名称，机构描述，学习人数等基本信息。具体表结构如表4-6所示。

表4-3 courseorg表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 大小 | 允许为空 | 是否主键 | 备注 |
| 1 | id | int | 11 | 否 | 是 | 机构ID |
| 2 | name | varchar | 50 | 否 | 否 | 机构名称 |
| 3 | desc | longtext |  | 否 | 否 | 机构描述 |
| 4 | tag | varchar | 10 | 否 | 否 | 机构标签 |
| 5 | category | varchar | 20 | 否 | 否 | 机构类别 |
| 6 | click\_nums | int | 11 | 否 | 否 | 点击数 |
| 7 | fav\_nums | int | 11 | 否 | 否 | 收藏数 |
| 8 | image | varchar | 100 | 否 | 否 | 封面图 |
| 9 | address | varchar | 150 | 否 | 否 | 地址 |
| 10 | students | int | 11 | 否 | 否 | 学习人数 |
| 11 | course\_nums | int | 11 | 否 | 否 | 课程数 |
| 12 | add\_time | varchar | 200 | 否 | 否 | 添加时间 |

teacher表中主要存储讲师信息，有讲师名称以及工作年限，教学特点等。具体表结构如图4-7所示。

表4-7 teacher表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 大小 | 允许为空 | 是否主键 | 备注 |
| 1 | id | int | 11 | 否 | 是 | 讲师ID |
| 2 | name | varchar | 50 | 否 | 否 | 讲师名称 |
| 3 | work\_years | int | 11 | 否 | 否 | 工作年限 |
| 4 | work\_company | varchar | 50 | 否 | 否 | 就职公司 |
| 5 | work\_position | varchar | 50 | 否 | 否 | 公司职位 |
| 6 | points | varchar | 50 | 否 | 否 | 教学特点 |
| 7 | age | int | 11 | 否 | 否 | 年龄 |
| 8 | click\_nums | int | 11 | 否 | 否 | 点击数 |
| 9 | fav\_nums | int | 11 | 否 | 否 | 收藏数 |
| 10 | image | varchar | 100 | 否 | 否 | 头像 |
| 11 | add\_time | datetime | 6 | 否 | 否 | 添加时间 |
| 12 | org\_id | int | 11 | 否 | 否 | 机构外键 |

lesson表主要存储章节信息。具体表结构如表 4-8所示。

表4-8 lesson表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 大小 | 允许为空 | 是否主键 | 备注 |
| 1 | id | int | 11 | 否 | 是 | 章节ID |
| 2 | name | varchar | 100 | 否 | 否 | 章节名称 |
| 3 | add\_time | datetime | 6 | 否 | 否 | 添加时间 |
| 4  5 | course\_id | int | 11 | 否 | 否 | 课程外键 |

video表主要存储视频信息。具体表结构如表4-9所示。

表4-9 video表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 大小 | 允许为空 | 是否主键 | 备注 |
| 1 | id | int | 11 | 否 | 是 | 视频ID |
| 2 | name | varchar | 100 | 否 | 否 | 章节名称 |
| 3 | add\_time | datetime | 6 | 否 | 否 | 添加时间 |
| 4 | url | varchar | 200 | 否 | 否 | 访问地址 |
| 5 | learn\_times | datetime | 6 | 否 | 否 | 视频时长 |
| 6  5 | lesson\_id | int | 11 | 否 | 否 | 章节外键 |

第5章 系统详细设计及实现

本系统分为两个部分：网页前端界面和后台管理系统。网页界面包括用户注册、用户登录、个人中心、课程学习、用户收藏等功能。系统后台管理系统包括用户管理、课程管理、机构管理、讲师管理、网页前端管理等功能。

5.1 网页前端设计

5.1.1 登录功能

前端登录页面，此界面实现用户登录功能，登录时需与注册时用邮箱，密码一致方可登录。登录功能如图5-1所示。



图5-1 登录界面

实现登录功能核心代码如下：

class LoginView(View):

def get(self, request):

return render(request, "login.html", {})

def post(self, request):

login\_form = LoginForm(request.POST)

if login\_form.is\_valid():

user\_name = request.POST.get("username", "")

pass\_word = request.POST.get("password", "")

user = authenticate(username=user\_name, password=pass\_word)

if user is not None:

if user.is\_active:

login(request, user)

return HttpResponseRedirect(reverse("index"))

else:

return render(request, "login.html", {"msg": "用户名未激活！"})

else:

return render(request, "login.html", {"msg": "用户名或密码错误！"})

else:

return render(request, "login.html", {"login\_form":login\_form})

5.1.2 注册功能

在用户注册界面，此界面也采用了系统经典的简约风格设计，给用户一种清新、干净的感觉。用户填写邮箱，密码，验证码等信息即可实现注册功能。增添邮箱验证功能，提高系统安全性和注册功能的可行性和易操作性。注册功能如图5-2所示。



图5-2 注册界面

实现注册功能核心代码如下：

class RegisterView(View):

def get(self, request):

register\_form = RegisterForm()

return render(request, "register.html", {'register\_form':register\_form})

def post(self, request):

register\_form = RegisterForm(request.POST)

if register\_form.is\_valid():

user\_name = request.POST.get("email", "")

if UserProfile.objects.filter(email=user\_name):

return render(request, "register.html", {"register\_form":register\_form, "msg": "用户已经存在"})

pass\_word = request.POST.get("password", "")

user\_profile = UserProfile()

user\_profile.username = user\_name

user\_profile.email = user\_name

user\_profile.is\_active = False

user\_profile.password = make\_password(pass\_word)

user\_profile.save()

#写入欢迎注册消息

user\_message = UserMessage()

user\_message.user=user\_profile.id

user\_message.message = "欢迎注册MOOC"

user\_message.save()

send\_register\_email(user\_name, "register")

return render(request, "login.html")

else:

return render(request, "register.html",{"register\_form":register\_form})

5.1.3 个人中心功能

此界面实现用户个人中心功能，包括用户信息管理，用户课程管理，用户收藏管理等功能。在本界面下用户可以进行自己的基本信息维护，也可以修改密码，可以查看用户学习过的课程以及收藏过的课程、机构、讲师。如图5-3所示。

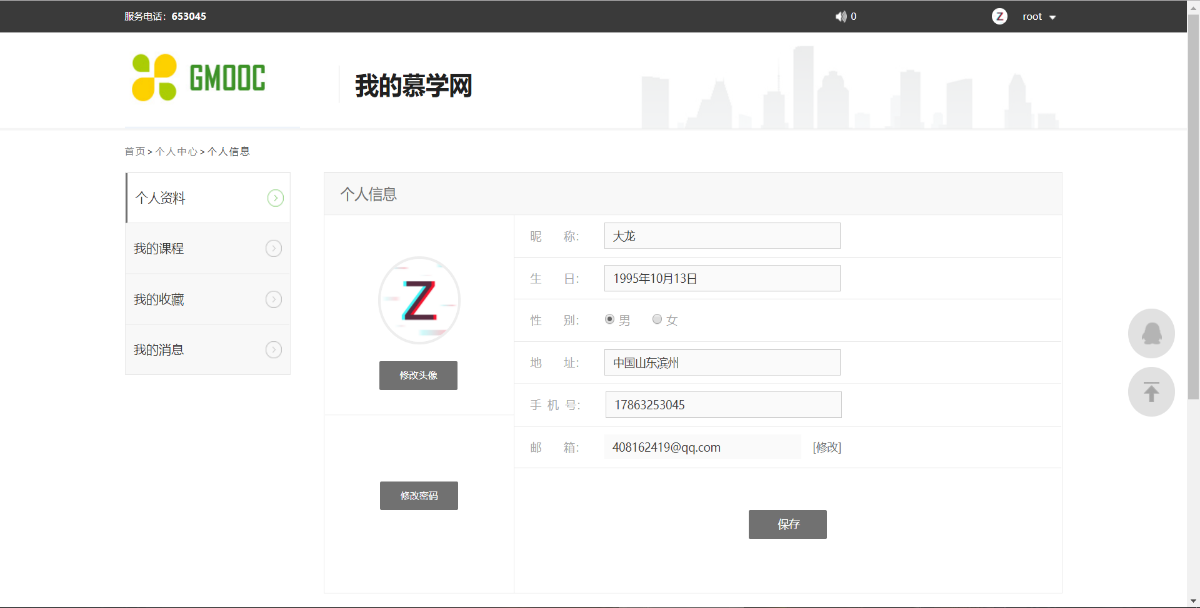


图5-3户中心主界面

5.1.4 课程学习功能

此界面实现课程学习功能，在课程详情页，用户只需要点击“开始学习”按钮，用户便可以进行该课程的学习。进入课程学习页面后，用户可以选择章节进行学习，也了解了该课程的详细信息。学习功能如图5-4所示。

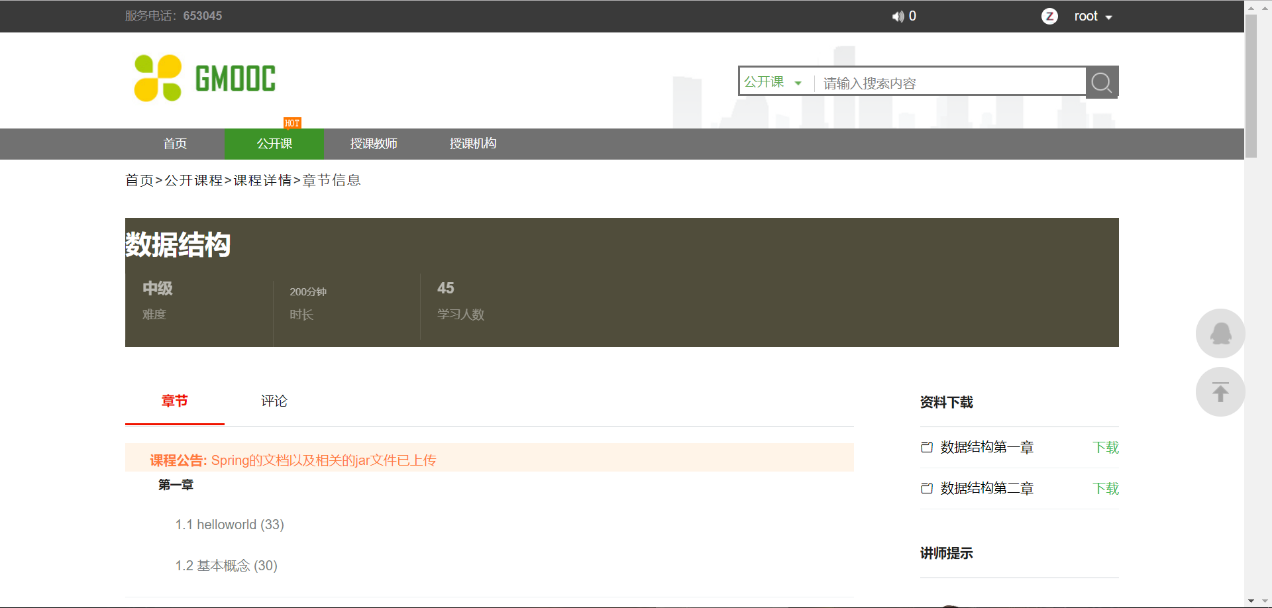


图5-4 课程学习界面

课程学习功能核心代码：

class CourseInfoView(LoginRequiredMixin, View):

#课程章节信息

def get(self, request, course\_id):

course = Course.objects.get(id=int(course\_id))

course.students +=1

course.save()

#查询用户是否已经关联了该课程

user\_courses = UserCourse.objects.filter(user=request.user, course=course)

if not user\_courses:

user\_course = UserCourse(user=request.user, course=course)

user\_course.save()

user\_courses = UserCourse.objects.filter(course=course)

user\_ids = [user\_course.user.id for user\_course in user\_courses]

all\_user\_courses = UserCourse.objects.filter(user\_id\_\_in=user\_ids)

# 去除所有课程id

course\_ids = [user\_course.course.id for user\_course in all\_user\_courses]

# 获取学过该课的用户还学过的课程

relate\_courses = Course.objects.filter(id\_\_in=course\_ids).order\_by("-click\_nums")[:5]

all\_resource = CourseResource.objects.filter(course=course)

return render(request, "course-video.html", {

'course': course,

'course\_resources':all\_resource,

'relate\_courses':relate\_courses,

})

class CourseDetailView(View):

"""课程详情页"""

def get(self, request, course\_id):

course = Course.objects.get(id=int(course\_id))

#增加课程点击数

course.click\_nums += 1

course.save()

#判断是否收藏

has\_fav\_course = False

has\_fav\_org = False

if request.user.is\_authenticated():

if UserFavorite.objects.filter(user=request.user, fav\_id=course.id, fav\_type=1):

has\_fav\_course = True

if UserFavorite.objects.filter(user=request.user, fav\_id=course.course\_org.id, fav\_type=2):

has\_fav\_org =True

tag = course.tag

if tag:

relate\_courses = Course.objects.filter(tag=tag)[:1]

else:

relate\_courses = []

return render(request, "course-detail.html",{

'course':course,

'relate\_courses':relate\_courses,

'has\_fav\_course':has\_fav\_course,

'has\_fav\_org':has\_fav\_org,

})

5.1.5 课程评论功能

此界面实现课程评论功能，用户只要学习过该课程，就可以对该课程进行评论，课程详情页面可以显示所有用户对课程的评论。课程评论界面如图5-5所示。

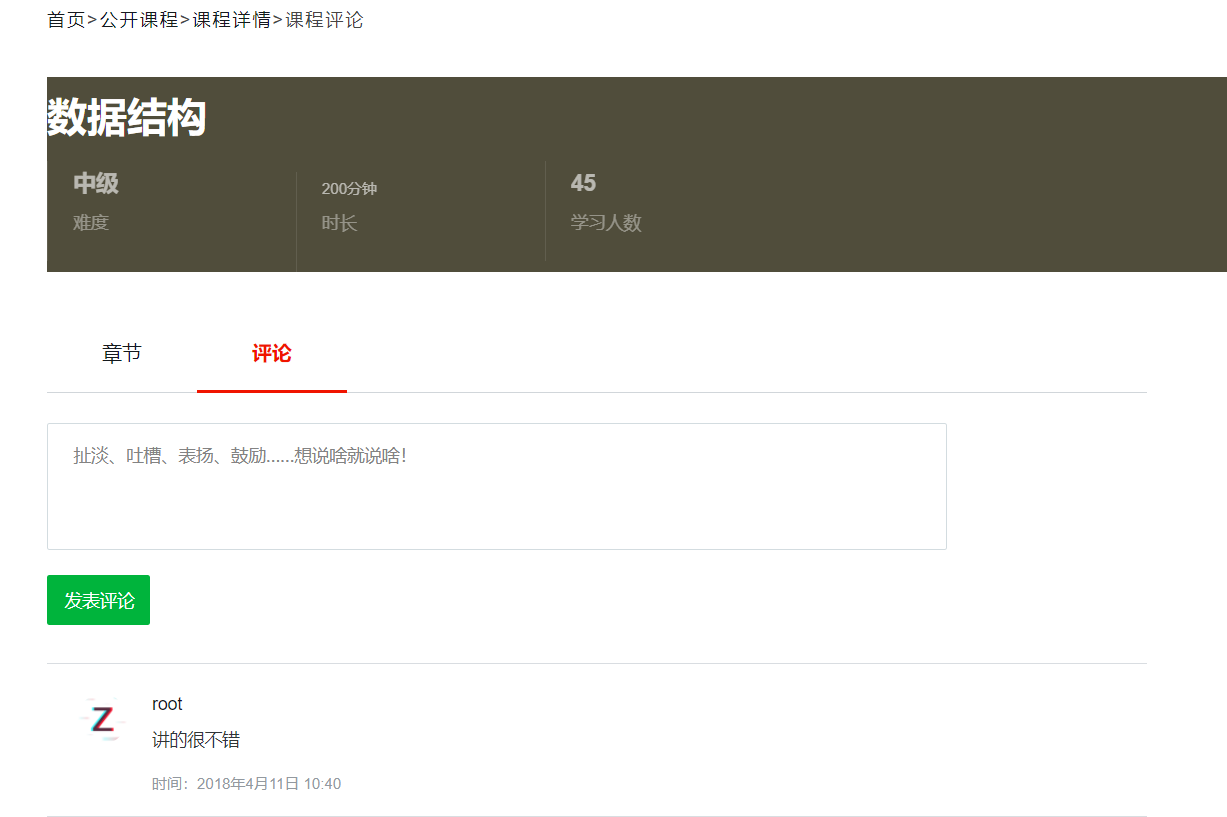


图5-5 课程评论界面

课程评论功能核心代码如下：

class CommentsView(LoginRequiredMixin, View):

def get(self, request, course\_id):

course = Course.objects.get(id=int(course\_id))

user\_courses = UserCourse.objects.filter(course=course)

user\_ids = [user\_course.user.id for user\_course in user\_courses]

all\_user\_courses = UserCourse.objects.filter(user\_id\_\_in=user\_ids)

# 去除所有课程id

course\_ids = [user\_course.course.id for user\_course in all\_user\_courses]

# 获取学过该课的用户还学过的课程

relate\_courses=Course.objects.filter(id\_\_in=course\_ids).order\_by/

("-click\_nums")[:5]

all\_resource = CourseResource.objects.filter(course=course)

#all\_comments = CourseComments.objects.all()

all\_comments = CourseComments.objects.filter(course=course)

return render(request, "course-comment.html", {

'course': course,

'course\_resources': all\_resource,

'all\_comments':all\_comments,

'relate\_courses': relate\_courses,

})

class AddCommentsView(View):

#用户添加课程评论

def post(self, request):

if not request.user.is\_authenticated():

#判断用户登录状态

return HttpResponse('{"status":"fail","msg":"用户未登录"}', content\_type='application/json')

course\_id = request.POST.get("course\_id", 0)

comments = request.POST.get("comments", "")

if course\_id >0 and comments:

course\_comments = CourseComments()

course = Course.objects.get(id=int(course\_id))

course\_comments.course=course

course\_comments.comments=comments

course\_comments.user = request.user

course\_comments.save()

return HttpResponse('{"status":"success","msg":"添加成功"}', content\_type='application/json')

else:

return HttpResponse('{"status":"fail","msg":"添加失败"}', content\_type='application/json')

5.1.6 收藏功能

点击界面的收藏按钮，可以进行收藏功能。可以收藏相应的课程、机构、讲师。收藏功能界面如图5-6所示。



图5-6 收藏界面

收藏功能核心代码如下：

class AddFavView(View):

"""

用户收藏,用户取消收藏

"""

def post(self, request):

fav\_id = request.POST.get('fav\_id', 0)

fav\_type = request.POST.get('fav\_type', 0)

if not request.user.is\_authenticated():

#判断用户登录状态

return HttpResponse('{"status":"fail","msg":"用户未登录"}', content\_type='application/json')

exist\_records = UserFavorite.objects.filter(user=request.user, fav\_id=int(fav\_id), fav\_type=int(fav\_type))

if exist\_records:

#记录已经存在，则表示用户取消收藏

exist\_records.delete()

if int(fav\_type) == 1:

course = Course.objects.get(id=int(fav\_id))

course.fav\_nums -= 1

if course.fav\_nums < 0:

course.fav\_nums = 0

course.save()

elif int(fav\_type) == 2:

course\_org = CourseOrg.objects.get(id=int(fav\_id))

course\_org.fav\_nums -= 1

if course\_org.fav\_nums < 0:

course\_org.fav\_nums = 0

course\_org.save()

elif int(fav\_type) == 3:

teacher = Teacher.objects.get(id=int(fav\_id))

teacher.fav\_nums -= 1

if teacher.fav\_nums < 0:

teacher.fav\_nums = 0

teacher.save()

return HttpResponse('{"status":"fail","msg":"取消收藏"}', content\_type='application/json')

else:

user\_fav = UserFavorite()

if int(fav\_id) > 0 and int(fav\_type) > 0:

user\_fav.user = request.user

user\_fav.fav\_id = int(fav\_id)

user\_fav.fav\_type = int(fav\_type)

user\_fav.save()

if int(fav\_type) == 1:

course = Course.objects.get(id=int(fav\_id))

course.fav\_nums += 1

course.save()

elif int(fav\_type) == 2:

course\_org = CourseOrg.objects.get(id=int(fav\_id))

course\_org.fav\_nums += 1

course\_org.save()

elif int(fav\_type) == 3:

teacher = Teacher.objects.get(id=int(fav\_id))

teacher.fav\_nums += 1

teacher.save()

return HttpResponse('{"status":"success","msg":"已收藏"}', content\_type='application/json')

else:

return HttpResponse('{"status":"fail","msg":"收藏出错"}', content\_type='application/json')

<script type="text/javascript">

//收藏分享

function add\_fav(current\_elem, fav\_id, fav\_type){

$.ajax({

cache: false,

type: "POST",

url:"{% url "org:add\_fav" %}",

data:{'fav\_id':fav\_id, 'fav\_type':fav\_type},

async: true,

beforeSend:function(xhr, settings){

xhr.setRequestHeader("X-CSRFToken", "{{ csrf\_token }}");

},

success: function(data) {

if(data.status == 'fail'){

if(data.msg == '用户未登录'){

window.location.href="/login/";

}else{

alert(data.msg)

}

}else if(data.status == 'success'){

current\_elem.text(data.msg)

}

},

});

}

$('#jsLeftBtn').on('click', function(){

add\_fav($(this), {{ course.id }}, 1);

});

$('#jsRightBtn').on('click', function(){

add\_fav($(this), {{ course.course\_org.id }}, 2);

});

</script>

5.2 后台管理设计

后台管理端的操作，包括用户管理、管理员管理、产品管理和订单管理。由于本毕业设计开发采用Django框架，后台实现的功能主要靠框架实现。因为该框架的操作是基于数据库中数据表的，所以实现功能代码不再列出。

5.2.1 后台管理主要功能

后台管理实现的是管理员对慕课管理系统的相关信息进行查看和修改。管理员可以在后台系统管理界面可以对用户信息的查看和修改，对课程信息的增删改查，对机构信息的增删改查，对教师信息的增删改查。同时管理员可以对一些信息其他信息进行修改和删除，如用户学习的课程、网站首页信息。如图5-7所示。

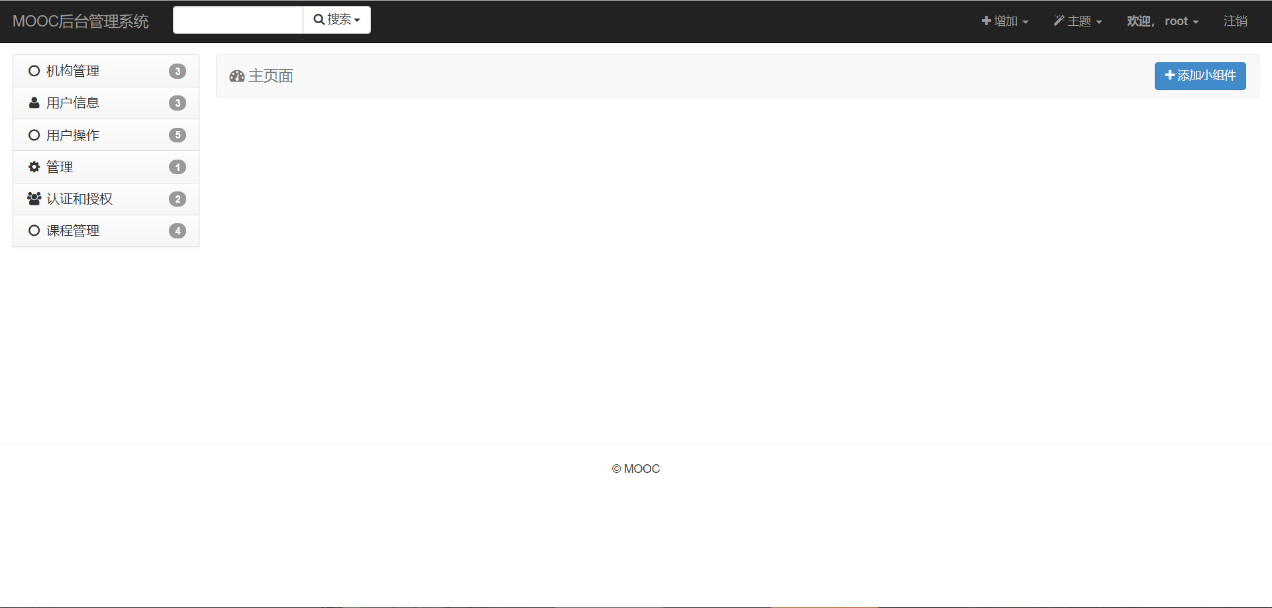


图5-7 后台功能主界面

5.2.2 信息添加功能

以添加课程信息为例。此功能实现的是管理员对课程信息添加操作。包括： 课程名称、课程描述、课程详情、课程等级、学习时间、学习人数、收藏人数、封面图、点击数、课程类别、课程标签、课程内容、添加时间、所属机构、授课讲师。添加课程信息界面如图5-8所示。

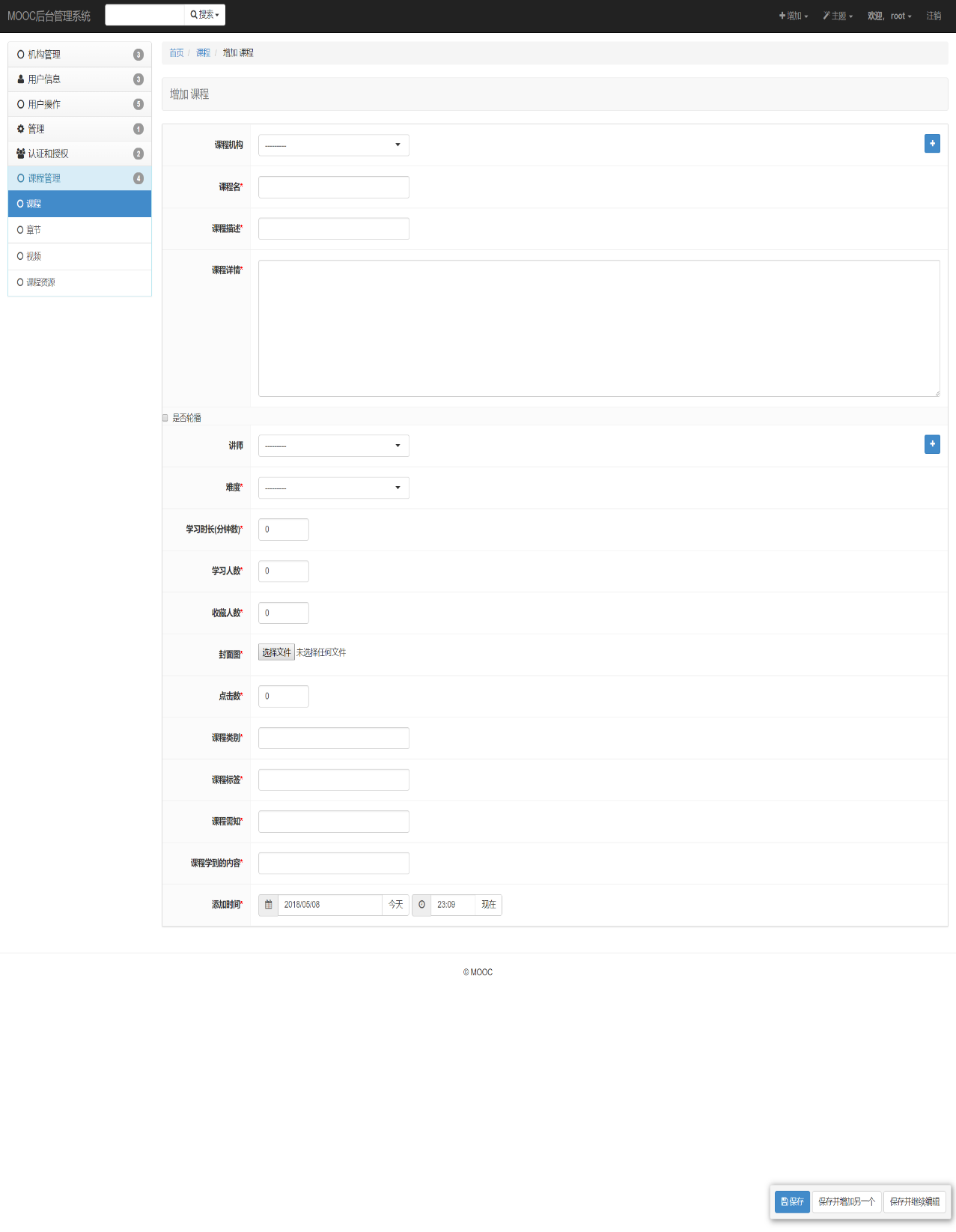


图5-8 课程信息添加

5.2.3 信息维护功能

以修改机构信息为例。此功能实现的是管理员对机构信息修改操作。包括：机构名称、机构描述、机构标签、机构类别、点击数、收藏数、封面图、机构地址、所在城市、学习人数、课程数、添加时间。修改机构信息界面如图5-9所示。

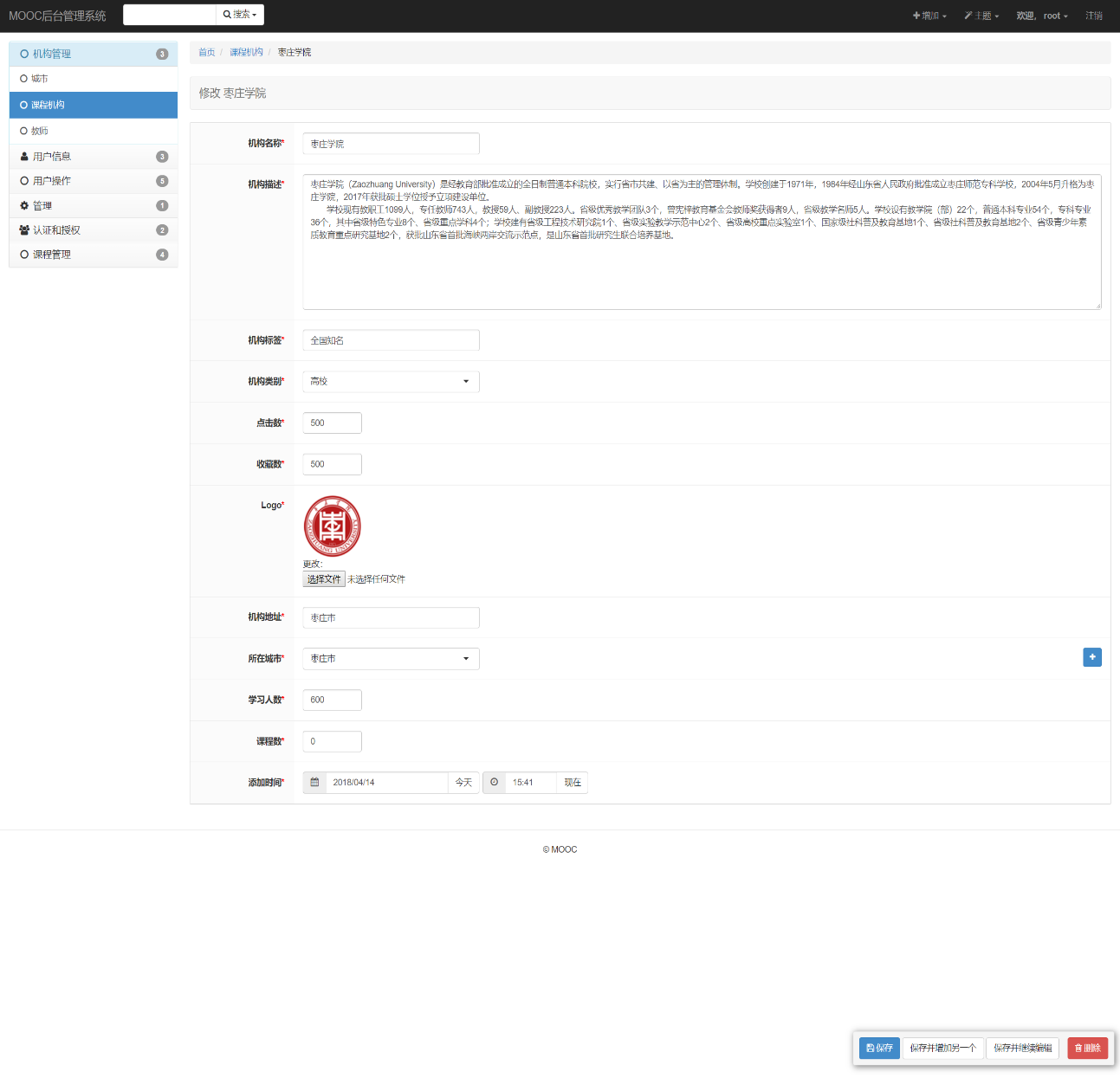


图5-9 机构信息修改界面

第6章 系统测试

6.1 系统测试目的

软件测试是软件交付至用户之前保证软件质量的重要方式，旨在尽可能早的发现错误并加以修正，从而检验阶段的成果是否能够达成目标。开发者在编写软件的过程中，难免会存在一些错误。因此，软件测试显得尤为关键，作为软件开发过程中的最后一步，决定着软件开发是否成功或开发的软件是否出色。因此，只有进行软件测试，才能知道软件中是否存在错误。

6.2 主要功能模块测试

本系统的主要功能是实现学生登录后的课程功能，所以测试的重点也放在这一部分，以确保它能实现系统的基本功能。其他相应功能，经过测试，均正常实现，本文主要测试系统的登录注册功能、学习功能以及收藏功能。

6.2.1 用户登录功能测试

登录/注册模块测试，如表6-1所示。

表6-1 测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项目 | 测试用例 | 测试结果 | 操作流程 |
| 用户登录 | 输入用户名：root  输入密码：root123456789 | 用户名或密码错误 | 点击登录按钮 |
| 用户登录 | 输入用户名：root  输入密码：root123456 | 登录成功！ | 点击登录按钮 |
| 用户注册 | 输入用户名：root  输入密码：root123456  输入错误的验证码 | 验证码错误！ | 点击注册按钮 |
| 用户注册 | 输入用户名：  输入密码：root123456  输入正确的验证码 | 您有未填写的字段！ | 点击注册按钮 |
| 用户注册 | 输入用户名：root  输入密码：123456  输入正确的验证码 | 用户已经存在！ | 点击登录按钮 |

当输入错误的登录信息时，提示登录错误，如图6-1所示。



图6-1 登录错误界面

用户注册成功，系统发送确认邮件，用户进行邮件确认后即可登录系统，如图6-2所示。



图6-2 注册成功界面

6.2.2 用户学习功能测试

用户学习测试，如表6-2所示。

表6-2 测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项目 | 测试用例 | 测试结果 | 操作流程 |
| 用户学习 | 未登录开始学习 | 跳转登录页面 | 点击开始学习按钮 |
| 用户学习 | 登录后开始学习 | 进入课程学习页面 | 点击开始学习按钮 |

用户开始学习后，该课程成为用户学习过的课程，如图6-3 所示。

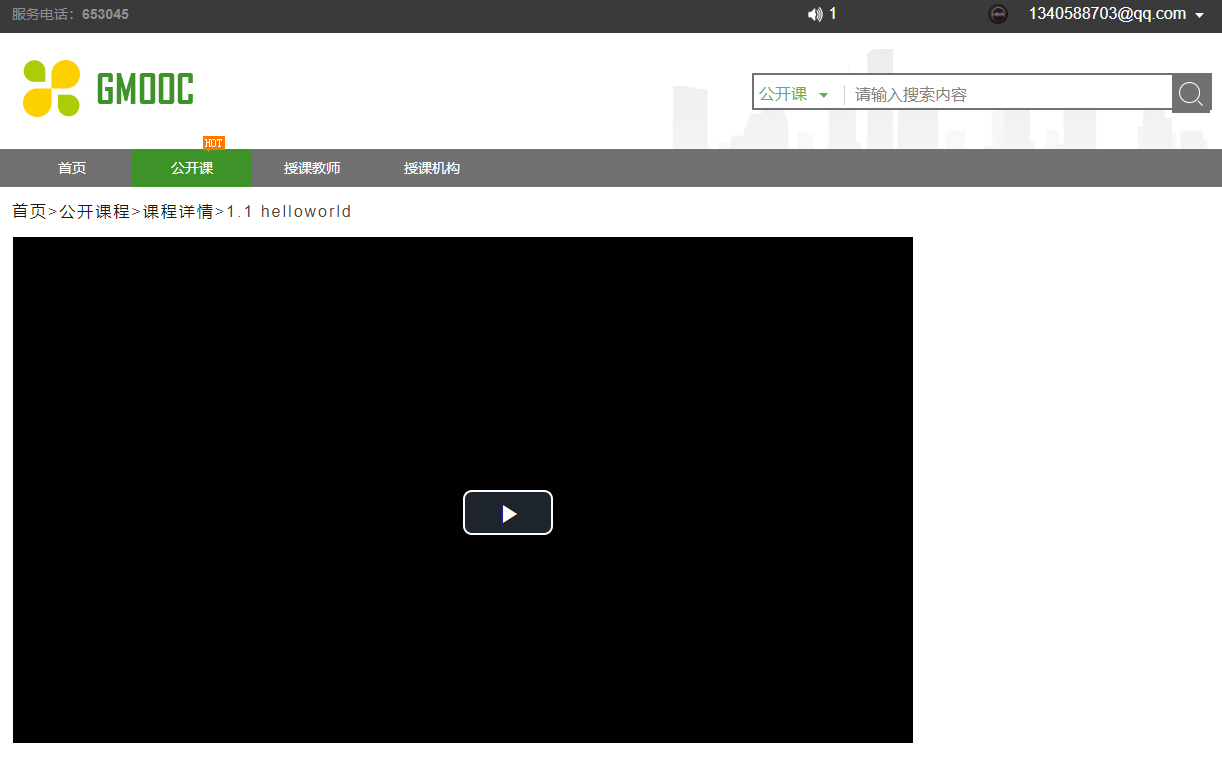


图6-3 用户学习界面

6.2.3 用户收藏功能测试

用户收藏功能测试，如表6-1所示。

表6-2 测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项目 | 测试用例 | 测试结果 | 操作流程 |
| 用户收藏 | 未登录课程收藏 | 跳转登录页面 | 点击收藏按钮 |
| 用户收藏 | 未登录讲师收藏 | 跳转登录页面 | 点击收藏按钮 |
| 用户收藏 | 未登录机构收藏 | 跳转登录页面 | 点击收藏按钮 |
| 用户收藏 | 登录后课程收藏 | 显示已收藏 | 点击收藏按钮 |
| 用户收藏 | 登录后讲师收藏 | 显示已收藏 | 点击收藏按钮 |
| 用户收藏 | 登录后机构收藏 | 显示已收藏 | 点击收藏按钮 |

用户收藏成功后，课程、讲师、机构与用户形成收藏关系，如图6-4所示。



图6-4 用户收藏界面

6.3 测试总结

通过对系统的各项功能进行测试，系统各项功能均可正常实现，系统的注册功能，登录功能，课程学习功能以及收藏功能均可运行。系统的后台的各项管理功能均可以正常实现。

第7章 系统总结与展望

经过近几个月的程序的编写及论文书写，毕业设计也随之进入收尾阶段，慕课管理系统的基本功能也逐个完成。本设计从功能上来说，已经基本实现了预期的用户登录、课程学习、相关收藏等系列功能。另外，系统还具备了课件下载功能，基本可以满足高校和企业的应用，使互联网教育拥有一个良好的展现平台，使教育能够惠及更多的人们。

本毕业设计从项目环境搭建开始，到Django中的app设计、models的编写、views的编写，深入理解了Django框架的开发使用过程。同时使用Django中的ORM框架，简化了数据库的设计流程。本毕业设计中使用的开源框架和工具大大加快了本项目的开发的速度。同时采用的B/S模式也使得该项目的跨平台能力有着显著的提升。

本毕业设计包括环境搭建，功能调试，每一步都锻炼了自己的动手能力。本项毕业设计，涵盖了专业中的诸多专业知识。在开发过程中，将这些知识用于实践，培养个人的动手能力与自主解决问题的能力，让我充分体会到自主探索的快乐。使本人认识到一个系统的全栈开发并非易事。而在设计中学习到的东西，更是一笔宝贵的财富，让本人受益终生。

当然本系统也存在许多不足之处需要后期来进行不断完善。后期拟对本毕业设计进行如下的完善和修改。

1.拟采用Nginx服务器进行上线，检查系统的易用性和可用性。

2.用户的权限问题。后期拟细化用户权限，对用户赋予部分权限。即讲师用户拥有相应权限、机构用户拥有相应权限、学校用户拥有相应权限，减轻管理员用户的负担。

3.网页前端界面。后期拟对网页前端界面进行响应式的适配，使得本毕业设计更好的适用于各个终端和平台，给用户的使用带来便捷。

4.对于数据库的设计可能存在某些不足之后，在后期可以进行相应的完善和补充。

在设计中还有些欠缺或考虑不周全的地方，但这次毕业设计的经历将对本人今后的学习和工作产生重大意义。

参考文献

1. 任友群,徐光涛,王美.信息化促进优质教育资源共享——系统科学的视角[J].开放教育研究,2013,(05):104-110.
2. 管佳,李奇涛.中国在线教育发展现状、趋势及经验借鉴[J].中国电化教育,2014,(331):62-65.
3. 汪琼.MOOCs改变传统教学[J].中国教育信息化, 2013,(19):26-28.
4. 何丽娴,甘淑,陈应跃.基于Python语言的空间数据处理[J].价值工程, 2014,(36):207-209.
5. 肖旻,陈行.基于Python语言编程特点及应用之探讨[J].电脑知识与技术,2014,(34):8177-8178.
6. 钱志远.基于Web的Python编程环境研究[J].数字技术与应用, 2016,(10):54.
7. 朱贇. Python语言的Web开发应用[J].电脑知识与技术, 2017,(32):95-96.
8. 郑岚. Python访问MySQL数据库[J].电脑编程技巧与维护,2010,(6):59-61.
9. 李云云.浅析B/S和C/S体系结构[J].科学之友,2011,(1):6-8.
10. 杨志庆.基于Django的Blog系统的开发与实现[J].机电一体化, 2013,(09):69-72.
11. 刘晓彬,高永平,单勇,黄敬仁.基于微信平台的课程管理系统设计[J]. 科技广场,2016,(5):186-189.
12. 王金龙,张静.基于JSP+MySQL的高校慕课(MOOC)教学系统设计[J].通讯世界,2017,(20):276-277.
13. 苟文博,于强.基于MySQL的数据管理系统设计与实现[J].电子设计工程, 2017,(06):62-65.
14. 张兴国.网上购物后台管理系统设计与开发[D].电子科技大学.2016.
15. 梁晓强,刘玲惠.企业网站后台管理系统的系统设计[J].才智,2012,(12):40.
16. Python Web Development with Django. J. Forcier,P. Bissex,W. Chun. Journal of Women s Health . 2008.
17. The Definitive Guide to MongoDB. Eelco Plugge,Tim Hawkins,Peter Membrey. Journal of Women s Health . 2010.
18. Python Programming Language. Van Rossum,G. USENIX Annual Technical Conference . 2007.
19. Programming python. MARK LUTZ.. 2011.
20. MongoDB:the definitive guide. Chodorow K . 2013.
21. Learning Website Development with Django. Hourieh A. . 2008.
22. Peter H. Welch,Herman W. Roebbers,Jan F. Broenink,Frederick R.M. Barnes,Carl G. Ritson,Adam T. Sampson,Gardiner S. Stiles,Brian Vinter,Waide B. Tristram,Karen L. Bradshaw. Hydra: A Python Framework for Parallel Computing[J]. Concurrent Systems Engineering Series,2009,67.

致 谢

大学四年就快要结束了，首先感谢枣庄学院信息科学与工程学院四年来对我的栽培，信息科学与工程学院为我提供了一个良好的学习环境，回顾四年来，感谢每一位教导过我的老师。大学四年以一篇毕业论文的形式结束，不禁心中感慨万千。

在毕业论文期间，我非常感谢指导老师对我的耐心、细心、全方位的指导，从选文到开题报告时期，指导老师都严格把关，循循善诱。从开题报告到毕业论文阶段，指导老师牺牲了很多时间，指导我在毕业论文写作中的一些具体问题，在此我向他表示我诚挚的谢意和崇高的敬意。同时也要感谢开源项目与工具的开发团队与作者，有了他们的项目和工具才使得本项目的开发更加简单、快速、准确。

几个月的紧张努力和学习，毕业论文结束了。在这篇论文的写作中，我深感自己对于计算机的道行不深，对于计算机的理解还十分欠缺，希望在之后我能继续不断弥补我现存的不足，不断进步。

最后，祝愿信息科学与工程学院蒸蒸日上，祝愿枣院的每一位学子前途光明，也祝愿所有老师们事业有成！