**毕业设计（论文）题目：MOOC平台设计模式的研究与实现**

**学 院：** 软件学院

**专 业：** 软件工程

**班 级：** 08211101

**姓 名：** 王璐

**指导教师：** 赵丰年

**校外指导教师：**



**毕业设计（论文）任务书**

**题目类别：** 毕业设计

**题目性质：** 工程设计

**毕业设计（论文）题目：** MOOC平台设计模式的研究与实现

学 院： 软件学院

专 业： 软件工程

班 级： 08211101

姓 名： 王璐

指导教师： 赵丰年

校外指导教师：

题目内容：

大规模网络开放课程(Massive Online Open Course，简称MOOC)产生于开放教育资源(Open Educational Resources，简称OER)不断运动发展的背景下。与OER一样，MOOC也是一种面向社会公众的免费教育形式。“大规模”意味着对同时参与学习的学习者数量不做限制，一门课程的学习者可以成百上千；“网络”意味着教与学的活动主要发生在网络环境下；“开放”意味着任何感兴趣的人可以注册学习，而且免费。MOOC是一种新型的学习和教学方法，它具有以下优点：易于使用；费用低廉，绝大数MOOC是免费的；覆盖的人群广；自主学习；学习资源丰富。本课题要求对目前主流MOOC平台（包括Coursera、Udacity、edX三大平台、可汗学院、网易公开课等）进行研究，分析总结MOOC平台的设计模式（design patterns，是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。使用设计模式是为了可重用代码、让代码更容易被他人理解、保证代码可靠性），最后使用自己熟悉的Web开发平台，实现一个简化版的MOOC平台。

任务要求：

对目前主流MOOC平台（包括Coursera、Udacity、edX三大平台、可汗学院、网易公开课等）进行研究，分析总结MOOC平台的设计模式和软件实现方法（重点研究游戏化设置、社交化设置等），最后使用自己熟悉的Web开发平台，实现一个简化版的MOOC平台。本系统的设计和开发要求完全按照规范化的软件项目来做，最后实现的系统既要有完整的注释完备的源代码，也要有所有合格软件产品要求的各种文档，并且实现的系统具有一定的可扩展性。

候选人要求：

1. 精通以下3种主流Web开发工具之一：PHP、.NET或Java；

2. 精通Web设计，熟练掌握HTML、CSS和Web设计模式；

3. 对教学和游戏有热情，熟悉Web上的各种机制（主要是教学机制、游戏机制和社交机制）。

具体任务：

1. 对MOOC平台进行研究和综述；

2. 总结MOOC平台开发方法和开发模式；

3. 设计一个符合主流需求的MOOC平台（至少能实现一门课程的大规模在线学习）；

5. 系统编码，实现网站；

6. 优化网站视觉效果和可用性；

7. 对整个网站进行测试；

8. 完成毕业设计论文；

9. 完成毕业论文英文资料翻译。

进度安排：

2014年12月20日~2015年2月15日：研究MOOC平台机制；

2015年2月15日~3月15日：总结MOOC平台开发方法和开发模式；

2015年3月15日~5月15日：设计MOOC平台，软件开发与实现；

2015年5月15日~5月30日：系统测试及功能测试

2015年4月15日~6月10日：撰写毕业论文，完成相关翻译

指导教师签字： 年 月 日

教学单位负责人签字： 年 月 日

责任教授签字： 年 月 日

**摘要**

随着计算机技术的发展和互联网的普及，教育行业也不再局限于传统的师-生课堂。2012年，MOOC出现在人们的视野里，并迅速的兴起和发展。比起传统课堂，MOOC支持大规模的学生同时在线学习，MOOC平台往往汇聚了来自全球顶尖大学的精品课程，且注重师-生交互和生-生交互。MOOC使得学习者能够足不出户，便接触到全球最顶尖的教学资源，和来自世界各地的学习者进行交流。短短几年时间里，MOOC从无到有，在全球范围内影响着传统教学方式，冲击着传统课堂。MOOC对教育的发展和变革的影响有着超出我们想象的可能性。

本文对现有的主流平台，如可汗学院，edX，Coursera，Udacity等进行分析，并着重研究其社交化和游戏化设计，总结出一套MOOC平台通用的设计模式，并依附于此设计模式，使用PHP语言和MySQL数据库，基于CodeIgniter框架进行开发，实现一个小型的，简化版MOOC平台。

**关键词：**MOOC；设计模式；PHP语言；CodeIgniter框架

**Abstract**

With the development of computer science and the popularity of the Internet, education industry is no longer limited to traditional “teacher to students” class. In 2012, MOOC appeared in people’s horizons, rose and developed rapidly. Compared to the traditional class, MOOC can support large-scale students studying online at the same time, and MOOC tends to gather high quality courses from top universities around the world. In addition to that, MOOC pays attention to the interaction between teachers and students, and the interaction between students. MOOC enables the learners to contact with the world's top teaching resources, and to interact with classmates from all over the world, even without going out of home. In just a few years, MOOC arises from scratch, impacting the traditional teaching method worldwide, and the traditional classroom. The possibility of MOOC's influence on the development of education can be beyond our imagination.

This paper focus on the design of current mainstream MOOC platforms, such as the khan academy, edX, Coursera and Udacity etc., social and gamification designs are studied primarily, and finally we come up with a set of design patterns for MOOC that’s used generally. With that set of design patterns, using PHP language and MySQL database, and based on the CodeIgniter framework, a small, simplified MOOC platform is built.

**Keywords：**MOOC; design pattern; PHP language; CodeIgniter framework

目录

[第1章 引言 1](#_Toc13112)

[1.1 什么是MOOC 1](#_Toc9013)

[1.2 MOOC的兴起与发展 1](#_Toc17614)

[1.3 课题研究的意义 2](#_Toc23943)

[1.4 本论文的主要工作 3](#_Toc25666)

[第2章 MOOC设计模式的研究 5](#_Toc16657)

[2.1 什么是设计模式 5](#_Toc27793)

[2.2 游戏化和社交化 5](#_Toc9591)

[2.2.1游戏化设计 5](#_Toc1288)

[2.2.2社交化设计 6](#_Toc21052)

[2.3 MOOC平台设计模式 6](#_Toc18137)

[2.3.1 课程进度 6](#_Toc13442)

[2.3.2 分数 10](#_Toc10356)

[2.3.3 勋章 13](#_Toc30602)

[2.3.4 信息广播-论坛 15](#_Toc9338)

[2.3.5 信息广播-分享 18](#_Toc15939)

[2.3.6 社会反馈 20](#_Toc394)

[2.3.7 状态保持 23](#_Toc28285)

[第3章 iMOOC平台的需求分析 27](#_Toc2362)

[3.1 系统总体需求分析 27](#_Toc17010)

[3.2 课程详情模块需求分析 27](#_Toc12271)

[3.2.1 课程详情模块的功能 27](#_Toc15292)

[3.2.2 用例图 27](#_Toc31867)

[3.3 讨论区模块需求分析 28](#_Toc11005)

[3.3.1 讨论区模块的功能 28](#_Toc13736)

[3.3.2 用例图 28](#_Toc22976)

[3.4 作业提交模块需求分析 29](#_Toc19738)

[3.4.1 作业提交模块的功能 29](#_Toc4186)

[3.4.2 用例图 29](#_Toc31269)

[3.5 考试模块需求分析 29](#_Toc16337)

[3.5.1 考试模块的功能 29](#_Toc19606)

[3.5.2 用例图 29](#_Toc6171)

[3.6 个人资料管理模块需求分析 30](#_Toc32382)

[3.6.1 个人资料管理模块的功能 30](#_Toc21809)

[3.6.2 用例图 30](#_Toc17781)

[3.7 留言模块需求分析 31](#_Toc8643)

[3.7.1 留言模块的功能 31](#_Toc9980)

[3.7.2 用例图 31](#_Toc9367)

[3.8 作业评分模块需求分析 32](#_Toc12515)

[3.8.1 作业评分模块的功能 32](#_Toc13784)

[3.8.2 用例图 32](#_Toc15835)

[3.9 排行榜模块需求分析 32](#_Toc25566)

[3.9.1 排行榜模块的功能 32](#_Toc28573)

[3.9.2 用例图 32](#_Toc27449)

[第4章 iMOOC平台的设计 33](#_Toc4942)

[4.1 系统的概要设计 33](#_Toc9215)

[4.2 系统模块的详细设计 33](#_Toc2573)

[4.2.1 课程详情模块的详细设计 33](#_Toc14002)

[4.2.2 讨论区模块的详细设计 34](#_Toc24260)

[4.2.3 排行榜模块的详细设计 35](#_Toc15677)

[4.2.4 作业提交模块的详细设计 35](#_Toc15286)

[4.2.5 作业评分模块的详细设计 36](#_Toc22734)

[4.2.6 考试模块的详细设计 37](#_Toc17770)

[4.2.7 个人资料管理模块的详细设计 37](#_Toc19359)

[4.2.8 留言模块的详细设计 37](#_Toc19923)

[4.3 系统的数据库设计 37](#_Toc24044)

[4.3.1 数据库总体设计 37](#_Toc10737)

[4.3.2 user表的设计 39](#_Toc20387)

[4.3.3 chat\_theme表的设计 40](#_Toc1339)

[4.3.4 chat\_detail表的设计 41](#_Toc25749)

[4.3.5 shomework表的设计 42](#_Toc10641)

[第5章 iMOOC平台的实现 43](#_Toc18225)

[5.1 网站开发环境 43](#_Toc10120)

[5.2 注册和登录模块的实现 43](#_Toc7693)

[5.3 课程详情模块的实现 44](#_Toc27214)

[5.4 讨论区模块的实现 48](#_Toc21198)

[5.5 排行榜模块的实现 51](#_Toc23416)

[5.6 作业提交模块的实现 51](#_Toc15656)

[5.7 作业评分模块的实现 53](#_Toc7802)

[5.8考试模块的实现 54](#_Toc15122)

[5.9个人资料管理模块的实现 55](#_Toc26772)

[5.10留言模块的实现 57](#_Toc18271)

[第6章 iMOOC平台的测试 59](#_Toc24955)

[第7章 总结 61](#_Toc27228)

[参考文献 63](#_Toc6671)

[附录 64](#_Toc31779)

[附录A video表的详细设计 64](#_Toc20942)

[附录B zan表的详细设计 64](#_Toc11972)

[附录C homework表的详细设计 65](#_Toc18672)

[附录D exam表的详细设计 65](#_Toc18034)

[附录E message表的详细设计 66](#_Toc3148)

[附录F score表的详细设计 66](#_Toc22609)

**第1章 引言**

**1.1 什么是MOOC**

MOOC是英文Massive Online Open Course的缩写，即大规模在线开放课程。“大规模”意味着巨大的数据量和学生人数；“开放”意味着课程资源向所有人开放，而不局限于特定用户；“在线”则意味着用户主要通过网络学习课程，几乎所有的教学环节均通过网络在线实现；而“课程”也不仅仅是狭义的知识内容，它包含了整个的教学过程，其中包括师-生和生-生交互。

MOOC使用了多元化的工具和资源。它整合了多种类的社交网络工具，并拥有多种形式的数字化资源。MOOC的课程易于大众学习和使用，因为它突破了传统课程在时间和空间上的限制，依靠互联网这个大平台使得全球的学习者都能够足不出户，在家学习到各顶尖高校的精品课程；MOOC拥有广泛的课程受众，它突破了传统课堂的人数限制，实现了大规模的学习者同时在线学习；同时，MOOC课程的参与是存在自主性的，学生可以自由加入课程，同样，也可以自由退出课程，因此，完成MOOC的学习内容需要学习者较强的自主学习能力和自我约束能力。

**1.2 MOOC的兴起与发展**

MOOC兴起于2012年，这一年也被称为“MOOC元年”。在这一年，数个美国顶尖大学及其教师陆续建立了MOOC平台，在网络上提供免费的课程资源，实现了美国顶尖大学优质课程资源的全球共享。此后，MOOC在互联网界引起极大关注，全球的学习者纷纷注册，加入MOOC的学习。其中Coursera，Udacity和edX是全球最具影响力的三个MOOC平台。其中Udacity在2012年2月由斯坦福大学计算机系教授Sebastian Thrun，David Stavens和Mike Sokolsky联合创办，有约400,000的注册用户；Coursera由斯坦福大学计算机系教授吴恩达和达芙妮.科勒（Daphne Koller）联合创办，和Udacity一样，它是一个盈利性的教育科技公司；edX则由麻省理工学院和哈佛大学共同创建，和前两个课程供应商不同，它是一个非盈利性的大规模在线开放课程平台。

而在中国，MOOC同样在逐步影响我们的教学方式，给传统课堂带来冲击。2013年5月，清华大学，香港大学和香港科技大学加入edX，成为该MOOC平台的首批亚洲高校成员；2013年2月，台湾大学加入Coursera，并推出了Coursera的全球首批中文MOOC课程，2013年7月，上海交通大学和复旦大学同样加入Coursera，成为Coursera的首批中国内地高校；而在2013年5月，北京大学加入了edX，随后又加入Coursera，此外，北京大学计划在未来5年时间里建立100门MOOC课程，对北大在校生和其他学生开放，在校生可以通过MOOC平台课程的学习获得等价于传统课堂的学分，其他学生则可以获得北大提供的证书。

除了加入各大国际主流MOOC平台，我们的各大高校也在尝试建立中国独有的MOOC平台。2013年6月，清华大学召开“大规模教育在线论坛”，拉开了中国MOOC研究和发展的序幕；2013年9月，上海交通大学举办“在线教育国际论坛”，国内外在线教育领域的权威集中探讨了在线教育所带来的革命性影响；2013年8月，华东师范大学组织成立了中小学MOOC联盟，专注于基础教育阶段的课堂革新，改革传统教育模式，促进了“翻转课堂”实施。

各大网络运营商也纷纷出动，抢占MOOC市场。2010年11月1日，中国领先的门户网站网易推出“全球名校视频公开课项目”，首批1200集课程上线，这些课程来自于哈佛大学、牛津大学、耶鲁大学等世界知名学府，内容涵盖人文、社会、艺术、金融等领域，用户可以免费在线观看这些课程。此外，网易公开课和Coursera，可汗学院展开合作，开发中文学习专区，消除了学习的语言障碍，降低了学习的门槛；网易云课堂也于2012年12月底正式上线，相比之下，该平台则更加注重实用技能的学习；2013年6月9日，优酷也与在线教育网站Udacity达成独家官方合作，成为全网首个也是现今全国唯一的Udacity课程发布渠道平台。

短短几年时间，MOOC伴随着互联网的发展，从无到有，在全球范围内影响着教学方式，冲击着传统课堂。人工智能和信息技术的飞速发展，使得MOOC对教育的发展和变革的影响有了超出我们想象的可能性。

**1.3 课题研究的意义**

随着互联网技术的发展和网络的普及，人们的生活变得和计算机息息相关。在越来越多的影响人们的娱乐，出行，医疗等方面的同时，网络在教育领域也产生了不可忽视的影响。在教育资源方面，人们开始不再满足于学习本校仅有的教学资源，而开始通过网络去主动学习，主动探索；在学习方式方面，人们也不再安于传统课堂“先学后练”的教学流程，翻转课堂开始被越来越多的应用；同时，人们也逐渐摆脱了“死读书”的传统，游戏化和社交化与教育的结合使得学习的过程增加了趣味性，和全球学习者的交流使得学生的知识面被无限的拓宽。

大规模在线开放课程（Massive Online Open Course，简称MOOC）则很好的满足了现今人们对教育的需求。MOOC平台“大规模”和“开放”的特点使它往往聚集了全球名校的优质教学资源，使学生可以足不出户而知天下事；而巨大的学生人数使得学员可以通过网络和全球的学习者产生交流，使得学习问题的解决更加快速，准确，同时增加了学生参与问题讨论的积极性；“在线”的特点消除了教学资源的地域限制，也降低了课程学习的门槛，只要是对该领域感兴趣的同学，都可以在线完成该课程的学习。

因此，伴随着现今的教育潮流，对MOOC的设计和实现进行深入的研究，充分发挥MOOC平台在教育领域的优势作用是有着巨大意义的。

**1.4 本论文的主要工作**

本论文紧紧围绕“MOOC平台设计模式的研究与实现”这一课题，对主流的MOOC平台，如Udacity，Coursera，edX，可汗学院等的设计模式和软件实现方法进行研究，并重点研究其游戏化和社交化设置，总结出一套MOOC平台通用的设计模式，并结合该套设计模式，使用PHP语言开发一个简化的小型MOOC平台，实现“中国古代建筑艺术”这门课程的在线学习。论文结构如下：

第 1 章 引言，重点对MOOC的兴起及发展进行分析，并阐述了课题研究的意义。

第 2 章 MOOC平台设计模式的研究，研究主流MOOC平台，总结出一套MOOC通用的设计模式。

第 3 章 iMOOC平台的需求分析，介绍了本文实现的简化版MOOC平台——iMOOC平台的功能需求。

第 4 章 iMOOC平台的设计，介绍了iMOOC平台的概要设计，各模块详细设计和数据库设计。

第 5 章 iMOOC平台的实现，介绍了iMOOC平台各模块的实现方法。

第 6 章 iMOOC平台的测试，测试iMOOC平台各模块功能，确保系统正常运行。

第 7 章 总结，总结本论文工作。

**第2章 MOOC设计模式的研究**

**2.1 什么是设计模式**

设计模式（design patterns）是一套可以反复使用的，为多数人所知晓的，经过分类编目的，前人代码设计经验的总结。每个模式都针对一个在我们的开发过程中会反复出现的问题，使用设计模式，我们可以避免一次又一次的重新进行相同问题解决方案的研究，增加了代码的可重用性。同时，设计模式的存在使得代码更容易被开发者之外的人理解，提升了代码的可读性，也增加了系统的可维护性与可扩展性。设计模式的存在使得软件的编码过程真正的工程化，是整个软件工程体系的基石。

设计模式有助于我们对框架的理解。设计框架通常是一组相互协作的类，构成了某类特定软件的可复用的设计。它通常定义了应用程序的整体架构，类和对象的关系等参数，使开发者能专注于应用本身的特定细节，而不是在应用中出现的共同的设计决策，强调复用性。因此，设计模式在成熟的框架中不可或缺，在熟悉设计模式之后，开发者通常能够迅速掌握框架的结构。

设计模式使用的根本目的是提高代码的可重用性与系统的可维护性，为了达到这些目的，设计模式需要遵循一些设计原则，如开闭原则（Open Closed Principle，OCP）、里氏代换原则（Liskov Substitution Principle，LSP）、依赖倒转原则（Dependency Inversion Principle，DIP）、接口隔离原则（Interface Segregation Principle，ISP）等。

**2.2 游戏化和社交化**

**2.2.1游戏化设计**

游戏化设计，是指运用游戏元素和游戏设计的技巧，来解决非游戏问题，包括使用分数机制，排行榜，勋章等。在MOOC中，游戏化的设计可以减轻学生的学习压力，增加学生的学习动力，激发学生探索课程知识的意愿，使学生更愿意去主动的学习知识。

使用游戏化设计并不会保证用户参与到游戏化的部分中，Hamari (2012)提出，简单的提供游戏环境并不表示用户会主动的去参与。例如，在玩象棋时，人们并没有得到“游戏感”。相反的，在非游戏的环境中，人们可能会有游戏化的体验，例如参与“证券市场”等。因此，并不能仅仅简单的建立游戏化环境，因为不知道用户是否会愿意主动的参与其中，且如果用户参与了，是否会按照设计者期待的方式去做。错误的滥用游戏化设计很有可能会适得其反，鼓励了设计者不希望的行为。

在开发过程中，使用游戏化设计需要设计者不仅仅停留于和用户的浅层互动，例如游戏机制里的“等级”，“勋章”等，它们往往会很快失去它们的意义。设计者需要对游戏化设计模式的内在动力有更深刻的理解。游戏化设计应该鼓励用户去探索系统，主动学习。可汗学院中的“能量积分”机制和勋章体系等都是MOOC平台现有的典型游戏化设计。

**2.2.2社交化设计**

社交化设计为用户提供了与他人交流的机会，满足了用户的和社会沟通的需求。社交化的设计利用交互性来增加用户粘性，并帮助用户获得大量的社会反馈。设计者在自己的软件平台中加入社交化设计依旧是很流行的，打开自己手机的应用商店，几乎所有的应用都提供了到各大社交平台的分享功能，每个应用都有用户的评分和评论的展示。用户绝对不是自己一个人在使用软件，而是在和全球其他的软件用户交流，分享。各种社交化平台也在如火如荼的发展，国外的Twitter，FaceBook，Instagram，国内的微博，微信，豆瓣，知乎，为社交化的设计提供了良好的基础。

对MOOC平台来说，比起之前经常使用通讯录等设计，建立用户之间的实时联系的社交化机制，MOOC更常使用的是在用户间建立异步联系，例如借助多种多样的第三方平台实现课程信息和视频信息的分享，和建立拥有诸多子版块的课程讨论区以供用户进行知识的交流等。

需要注意的是，设计者不能强迫用户去社交化，也不能喧宾夺主，软件平台仍然需要注重自己本身的核心功能，也需要引导用户去关注自己使用该平台的核心目的。通常，在用户需要将信息传播给一个或更多的人的时候，或者软件需要利用“社交网络”中人们的口口相传来成长，来扩大影响力的时候，设计者才会使用社交化设计。

**2.3 MOOC平台设计模式**

**2.3.1 课程进度**

描述：

在MOOC平台中，加入某门课程后，系统通常会列出完成本门课程学习所需要完成的任务，就像一个“课程进度条”一样，用户通过“课程进度条”的指引依次完成学习任务。

课程的进度条列出了学生在本门课程的学习中需要完成的学习任务，学生只需要根据进度条的指引依次完成任务，而不需要自己考虑下一步要进行的学习。进度条通常是动态的，以图形化的形式实时呈现已完成和未完成的学习任务的分布情况。这样的学习方式增加了学生的自主性，激发了学生对知识的好奇心。完成任务项后进度条的变化使得学生获得成就感，因而有了更强的学习动机。

课程的进度条实际赋予了用户一个隐含的目标：用户需要逐个完成任务，以达到进度条的末端是一种“梯度目标”的模式。它会即时反馈给用户用户的学习情况，使用户对自己达成目标的程度有了更好的了解。在传统课堂或者普通的在线教育中，用户在离开很久之后再返回本门课程继续学习时，可能会忘记自己的学习进度，从而对本门课程产生消极心理，缺乏继续学习的动力；而进度条的存在使目标时刻直观存在于整个学习进程中，从而持续的提升了学习的积极性。

从另一个角度来看，除了已完成任务外，进度条当然也实时的提醒着用户尚未完成的任务。“蔡氏效应”(Zeigarnik effect)研究显示：比起已处理完的事情，人们往往对于尚未完成的事情印象更加深刻。进度条使得这种“未完成”更加直观的出现在学习者眼前，使得用户有更加强烈的完成学习的意识。

举例：

MOOC的进度控制会以多种多样的形式呈现给用户。进度控制可能针对整门课程，也可能针对课程中的某个课时，某次作业。

如图2-1为可汗学院Early Math课程的总体完成情况，它采用图形和数字方式显示了已完成的课程习题数占习题总数的百分比。其中各个级别习题的完成情况以饼状图方式显示，图中蓝色部分表示完成了占总任务量2%的一级习题；图2-2显示的是Udacity课程中某个课时的学习进度，以不同颜色表示不同的状态，浅蓝色表示已经学习完的部分，橘色表示正在学习的部分，灰色则表示还未学习的部分；图2-3为Coursera中某门课程的课程目录，绿色圆圈表示已经完成学习的部分，灰色圆圈表示未完成学习的部分。

在课程的进度条中，每个任务通常能在10分钟以内完成，这为学生提供了即时的反馈，学生可以很快的看到自己的进步，且大量的学习活动和进度队列可以让学生全身心的投入到学习中去。设计者也可以在进度条方面进行改进，同时强调已完成和未完成的任务数，同时注重任务梯度模型和莫氏效应，激励学生完成剩下的学习。

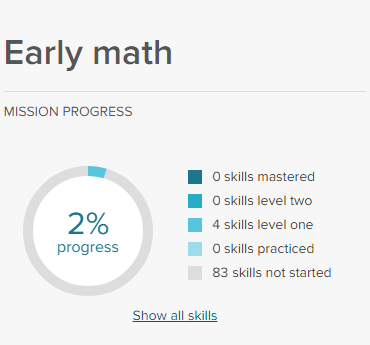


图2-1 可汗学院Early math课程的进度控制



图2-2 Udacity中某个课时的进度

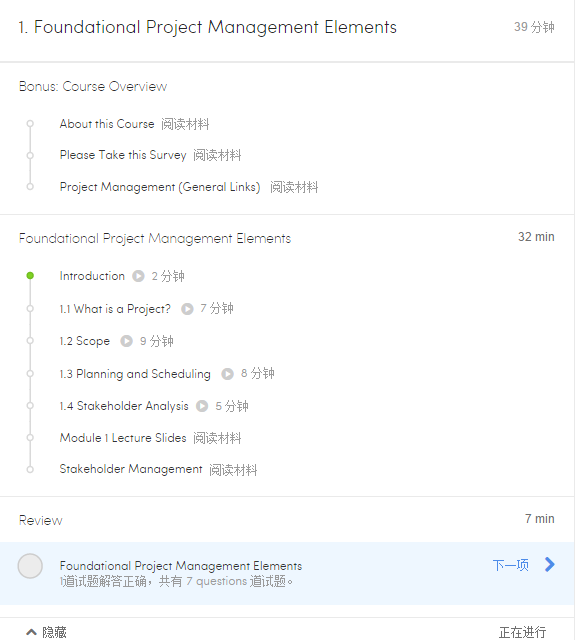


图2-3 Coursera课程目录的进度控制

实现：

以MOOC学习中的视频播放列表为例，在完成某学习任务后对该任务进行标记，并更新数据库。其实现方式如下

|  |
| --- |
| //建立数据表存储视频进度信息  function create\_table( ){  $sql = "create table process(  video\_name varchar(100) not null,  ......,  sign int(1) not null default 0 ) ";  mysql\_query($sql);  }  function process( ){  $curTime = $ video.currentTime;  $totalTime = $video.duration;//获取当前播放视频文件的进度和总时间  if($curTime ==$ totalTime){  goto $nextvideo.url;  update process set sign=1 where video\_name=$video.name;  }//当当前视频播放完时对当前视频进行标记，更新数据库中学习进度相关表并自动播放下一个  } |

**2.3.2 分数**

描述：

分数往往是一个量化的值，它是对用户特定行为的奖励，也可以用来表现用户成功度，评定用户等级。

分数通常定义为设计者在想要给予积极参与的用户反馈，或量化设计者对用户行为的满意度时所使用的模式。分数通常在用户行为易于量化时使用，如果不是必须以量化的形式呈现，也可以使用“点赞”等形式来体现。

分数几乎可以被用于任何离散或非离散的，可计数的数量问题上。分数可以以连续不断的方式呈现，就像学生的考试成绩，课程成绩一样；他们也可以以一种离散的方式呈现，例如1-5分的星级课程评分形式，或者是学生的学习情况，专业知识的掌握情况（初学者，中级学习者，专家级别学习者等）的体现。距离效应研究显示，当数字变大时，虽然绝对差并没有变化，但在人们的心理上，数字之间的差距却被压缩了。例如在买东西时，人们总会觉得6元到5元之间的降价差距是要大于56元到55元的。在商场进行促销活动时，6元降到5元的商品也会更多的被顾客购买。因此，分数的离散显示，例如评分等级，用户等级等，可能比仅仅使用连续数字的分数系统更加吸引用户。

举例：

在可汗学院中，使用分数机制构建了“能量积分(ENERGY POINTS)”系统。

相比学生对课程知识的掌握程度，能量积分的评估更侧重的是学生在学习过程中的努力程度。在学生的能量积分达到一定阈值时，又可以解锁系统特定的形象。

可汗学院的分数机制也存在一些问题。“能量积分”机制的建立是为了以物质激励的形式来刺激学生的学习，然而分数却可以通过完成一些几乎不需要任何努力的任务获得。使用分数机制可以使一些学生因为游戏化的特性而最大化的使用它，然而分数机制的使用同时也使学习进程处于“被游戏”的高风险之中。现在的可汗学院还没办法保证学生观看视频时的活跃度（例如使用嵌入式测验，在视频播放中间进行简单的重要知识点测试，学生只有答对问题才能继续观看下面的视频），学生可能会为了获得更多积分而打开很多标签页，却仅仅让视频自动播放，而不是自己认真观看，从而忽略了学习本门课程的真正目的；此外，学生也可能为了获得积分而去学习一些对自己没有帮助的简单课程，例如数学的四则运算，拼音等，之后通过回答大量简单的问题而获得积分。

可汗学院的另一个特点是在答题过程中提供了“问题提示”，且这个提示很可能会直接告诉用户答案，从而降低测验结果的准确性（如图2-4）。用户可能仅仅打开多个带有提示的问题标签页，就可以正确回答所有问题，而没有学到任何实际的知识。

分数其实暗含了对用户特定行为的物质刺激。在此处，分数刺激用户观看课程视频和正确的回答问题。然而分数却不能刺激用户学着去解决问题。就像做数学分析题一样，结果不是最重要的，重要的是得出结果前的推理过程。为了解决这个问题，可以使用更复杂的设置，例如迫使用户写出分解的计算过程，然而这种机制在可汗学院中还没有实现。

如图2-5，为可汗学院中的分数机制。在用户答完6道题并全部答对后，获得了700分的能量积分，其中300分因为100%的正确率获得，300分因为完成此项任务本身而获得，100分因为在任务中获得了一枚勋章而得到。



图2-4 可汗学院中的提示

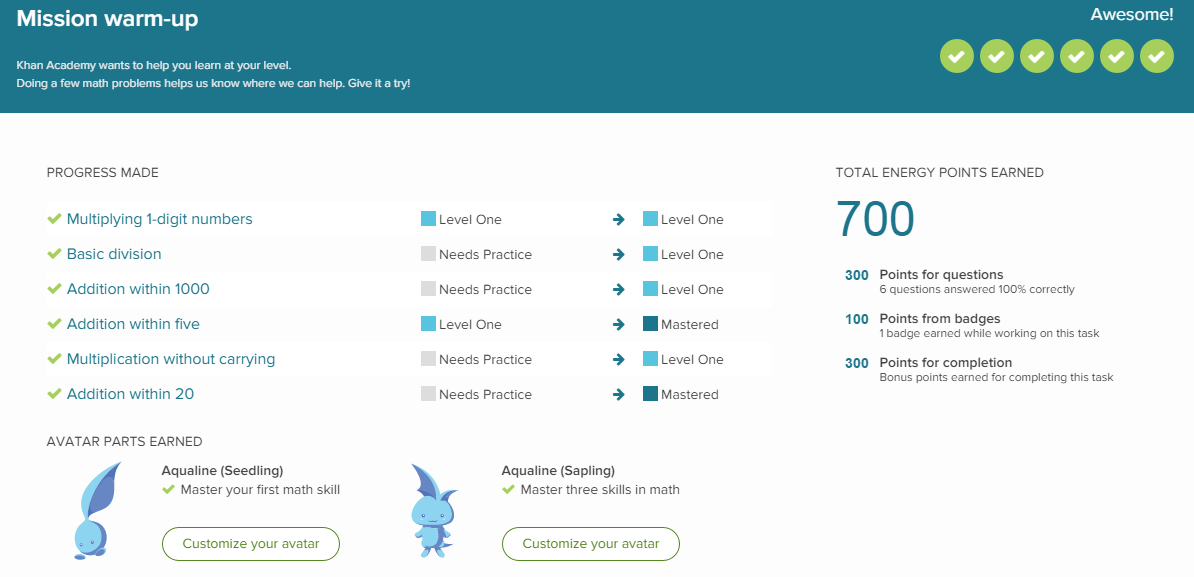


图2-5.可汗学院中的分数机制

实现：

分数机制的实现代码如下。

|  |
| --- |
| //建立用户表时创建分数字段记录用户分数  function createtable( ){  $sql= "create table user(  name varchar(100) not null,  ......,  score int(10) not null default 0 ) ";  mysql\_query($sql);  }  funcition score( ){  if($video.curTime == $video.duration){  update user set score+=n where name=$user.name;  }//如果观看完视频，更新用户表，对应用户加一定分数  else if($\_POST["submit"] =="提交作业"&& $correct>m%){  update user set score+=n where name=$user.name;  }//如果完成作业且正确率达到标准，对应用户加一定分数  else if($\_POST["submit"] =="发起讨论"||$\_POST["submit"] =="回复"){  update user set score+=n where name=$user.name;  }//用户在讨论区发表或回复主题帖时加一定分数  else if($\_POST["submit"] =="考试"){  update user set score+=examscore where name=$user.name;  }//加上考试分数  } |

**2.3.3 勋章**

描述：

用户因为特定行为而被授予勋章，通常以“成就”的形式体现。

举例：

在可汗学院中，勋章包含“挑战补丁（challenge patches）”，“黑洞勋章（black hole badges）”，“太阳勋章（sun badges）”，“地球勋章（earth badges）”，“月球勋章(moon badges)”，“陨石勋章（meteorite badges）”。每个勋章类中又分有若干不同作用的同系勋章。一些勋章可以转换为用户的能量分数，一些可以用来给予用户特定的权利。勋章可以用来奖励用户的知识掌握，例如用户在一次性正确回答出若干问题后授予相应的勋章；也有些勋章是在用户长时间的坚持之后授予的，例如在成为可汗学院用户两年后，会被授予相应的勋章。

用户获得的勋章通常是不对他人公开的，只有用户自己和课程教师可见。用户获得勋章的一系列行为以“动作流”的方式呈现在用户的勋章面板上。然而学生主页通常只允许学生展示5个勋章，这可能会打击学生收集勋章的积极性。比起收集所有勋章来说，学生可能会直接收集对自己意义最大的勋章，并在主页上展示他们，从而忽略了实际知识的掌握。

和分数一样，勋章也是以物质刺激来激励学生的，学生在进行了开发者所希望的额外活动后会获得相应的奖励。然而，很多勋章的设计并不好，他们可能并不会带给学生他们希望的奖励，或者获得勋章所需要完成的任务是无意义的，且过于繁琐的设计也可能会降低用户了解系统，使用系统的积极性。

如图2-6，展示了陨石勋章的种类及示例用户获得陨石勋章的情况。

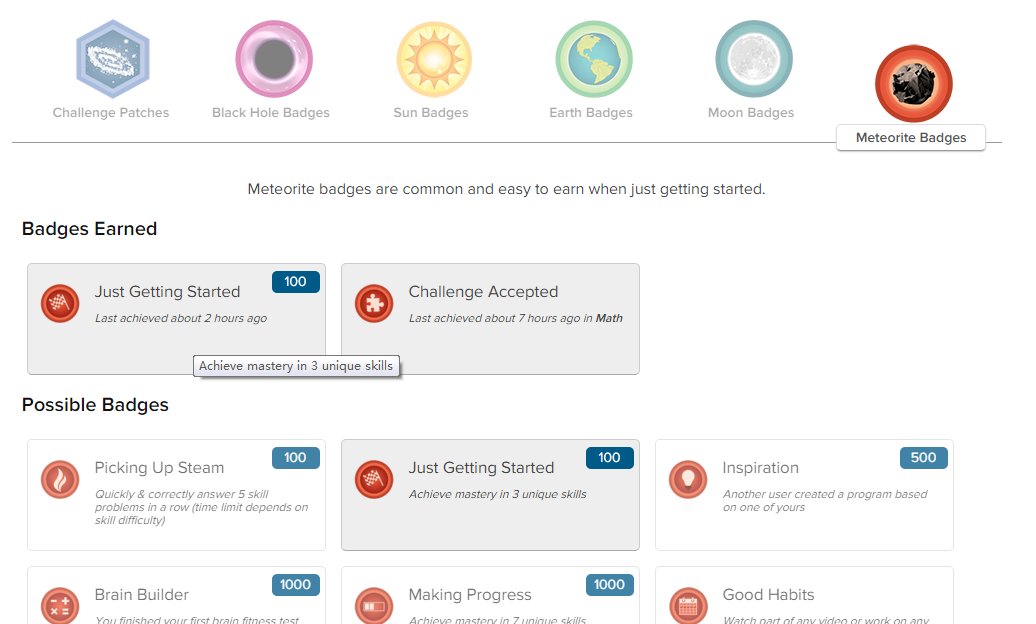


图2-6 可汗学院的勋章面板

实现：

用户在开始某门课程的学习之后，获得Just Getting Started勋章，示例代码如下。

|  |
| --- |
| //创建勋章面板，包含勋章种类，勋章数量和用户信息字段  function create\_table( ){  $sql=''create table badge(  badge\_type varchar(100) not null,  badge\_num int(5) not null,  user\_name varchar(100) not null  )'';  mysql\_query($sql);  }  function badge( ){  //当用户达到系统设置条件后  If($condition){  sql = ''select \* from badge where user\_name=$name'';  $res = mysql\_query($sql);  foreach($res as $item){  //授予相应类型勋章  If($item->badge\_type==''Just Getting Started''){  $item->badge\_num+=1;  break;  }  }  }  } |

**2.3.4 信息广播-论坛**

描述：

在论坛中，学生可以与他人分享，交流信息。通常当用户传播的信息需要被一个或多个人看见时，他们会使用信息广播。论坛即为一种广播的形式。通过论坛可以传播任何形式的媒介信息，包括文本，图片，音频，视频等，论坛是MOOC平台社交化的一个重要元素，他建立起了师生，生生之间的联系。

一门MOOC课程的论坛通常是树形结构的，良好的论坛结构的设置可以清晰的将相同主题的帖子汇集在一起，便于学习者快速解决问题，增加学习者访问讨论区的频率，也便于授课教师的教学管理，提高教学效率。通常，讨论区包含的模块有周话题区，技术问题区，作业讨论区和教学意见区。在实际应用中，设计者可以根据课程的实际需要来增加或删除论坛模块。

举例：

如图2-8中Coursera的论坛设置，它包含总讨论区（All Course Discussions），常规问题区（General Discussion），新人报到区（Meet and Greet），教学意见区（Content Issues），以及以章为单位的4个模块论坛。而图2-7中edX的论坛设置，则包含常规问题区（General），技术问题区（Technical Help），以及按课程目录结构划分的章节模块论坛。这些论坛的主要功能如下：

1.常规问题区：包含所有学生学习过程中遇到的课程相关的知识问题，可以是对课程中某个知识点的疑问，也可以是作业的交流讨论等。

2.技术问题区：包含学生和老师在使用该MOOC平台进行学习的过程中遇到的各种技术问题。在该讨论区中，学员和教师提出的技术疑问会得到平台开发团队的解答。

3.教学意见区：学生可以对整个教学过程提出自己的建议，说出自己独有的心得体会；学生可以对课程活动进行反馈，如教学进度的快慢，作业的难易程度等。教师可以定期登录该模块，收集教学建议，完善课程，使得该门课程的教学过程更加适合大多数学生。

4.模块论坛：通常根据课程的目录结构划分，每章可以作为一个子论坛，之后又可以划分为若干个课时的论坛。该讨论区将课程内容细化，学生可以根据自己的需要快速找到对应的讨论区，高效的论坛结构使得论坛访问频率增加，提高学生使用论坛的积极性，同时也便于教师的教学管理。

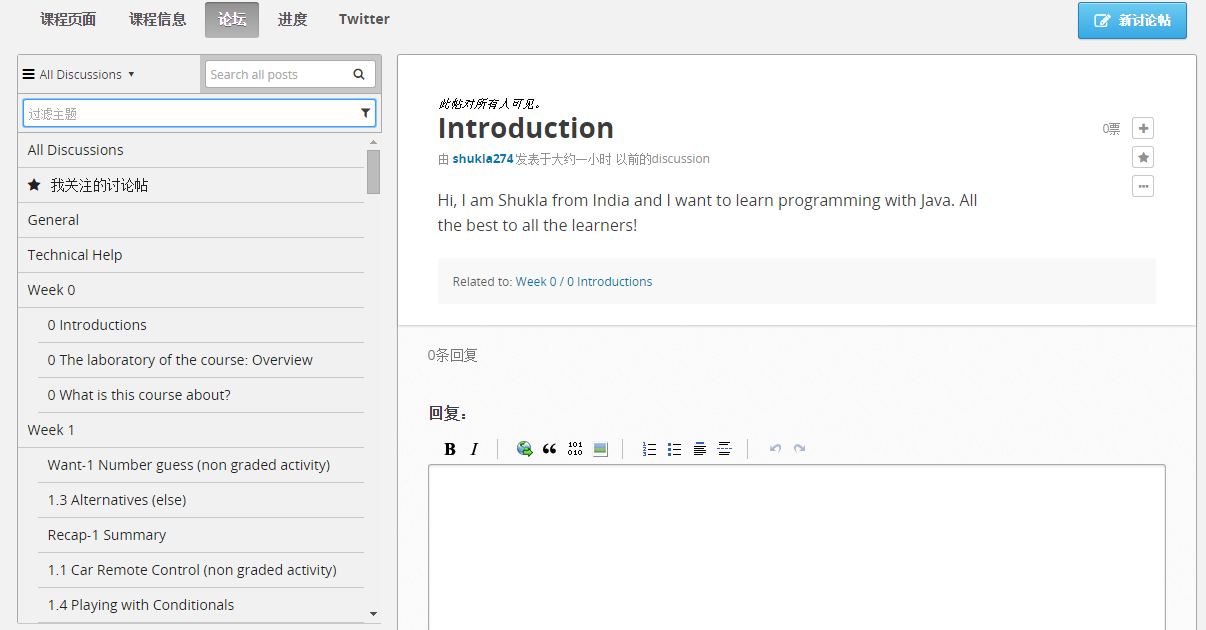


图2-7 edX论坛设置

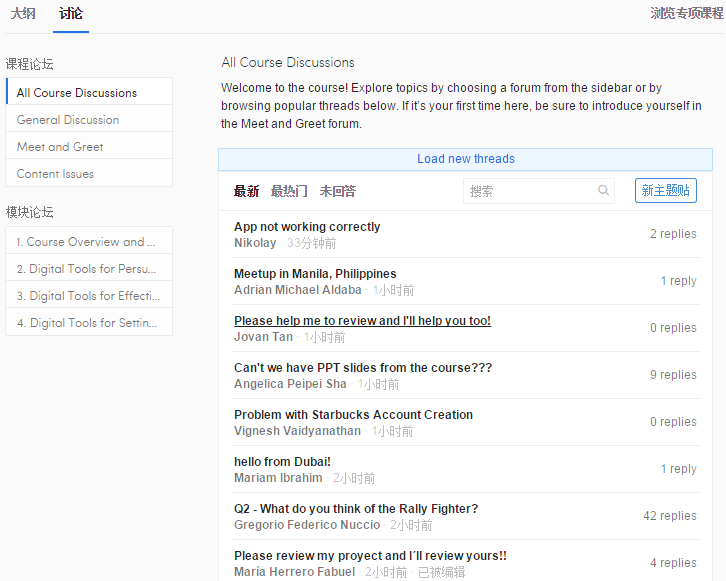


图2-8 coursera的论坛设置

实现：

论坛的数据需要存储在数据库中，建表代码如下。在发表主题帖时向chat\_theme表插入相应记录，在发表回复时向chat\_detail表插入相应记录。

|  |
| --- |
| //可以根据实际需求增加板块类型字段  function create\_chattheme( )  {  $sql = "CREATE TABLE IF NOT EXISTS `chat\_theme` (  `uid` int(10) unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,  `user` varchar(50) NOT NULL,  `theme` varchar(500) NOT NULL,  `content` text NOT NULL,  `time` datetime NOT NULL,  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci";  mysql\_query($sql);  }//建表chat\_theme存储主题，包含的字段有发帖人用户名，发帖题目，发帖内容，发帖时间  function create\_chatdetail( )  {  $sql = "CREATE TABLE IF NOT EXISTS `chat\_detail` (  `uid` int(10) unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,  `theme\_uid` varchar(500) NOT NULL,  `user` varchar(50) NOT NULL,  `content` text NOT NULL,  `time` datetime NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci";  mysql\_query($sql);  }//建表chat\_detail存储主题回复，包含的字段有主题的uid，发表回复的用户名，发表回复的内容，发表回复的时间 |

**2.3.5 信息广播-分享**

描述：

和论坛一样，分享机制也是信息广播的一种形式。在MOOC平台中，用户可以通过分享机制，将特定的课程信息通过Facebook，Twitter等发布。

举例：

如图2-9，为可汗学院的分享机制，在每个课时视频下方都设置了分享的下拉菜单，可以将视频信息发布到Twitter，FaceBook或者是邮件。如图2-10，为Udacity的分享机制，在课程名右下方，可以通过点击相应按钮将课程信息分享到FaceBook，Google+，Twitter。

实现：

分享功能可以通过调用js插件完成，例如使用JiaThis插件，代码如下。

|  |
| --- |
| <div class="jiathis\_style\_32x32" id="share">  <a class="jiathis\_button\_tsina"></a>  <a class="jiathis\_button\_tianya"></a>  <a class="jiathis\_button\_douban"></a>  <a class="jiathis\_button\_renren"></a>  </div>//分享到新浪微博，天涯论坛，豆瓣，人人  <script type="text/javascript" >  var jiathis\_config={  summary:"\*\*",  title:" ",  shortUrl:false,  hideMore:true,  siteNum:10,  }//分享的参数配置，默认分享当前界面url，分享内容为summary  </script>  <script type="text/javascript" src="http://v3.jiathis.com/code/jia.js"  charset="utf-8"></script> |

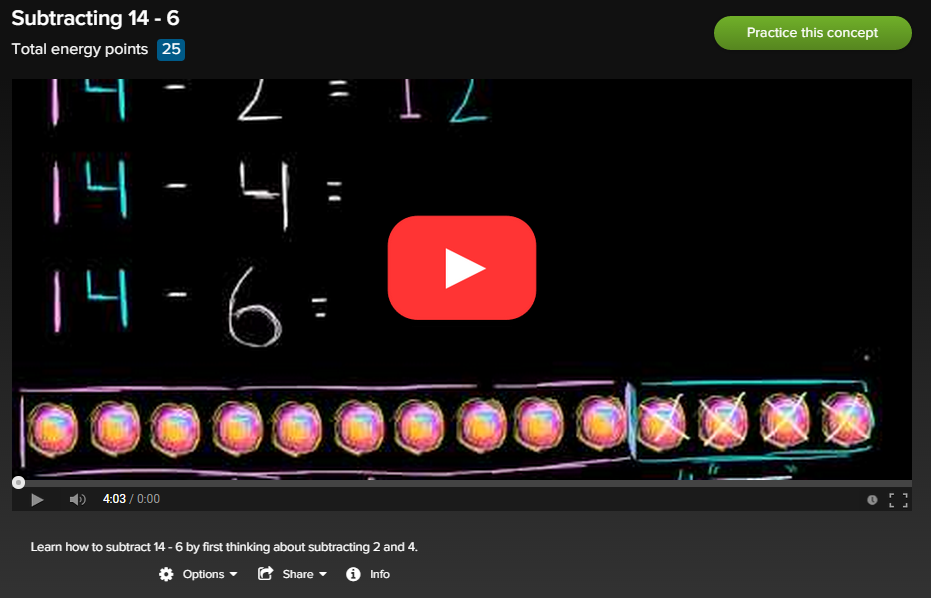


图2-9 可汗学院的分享机制

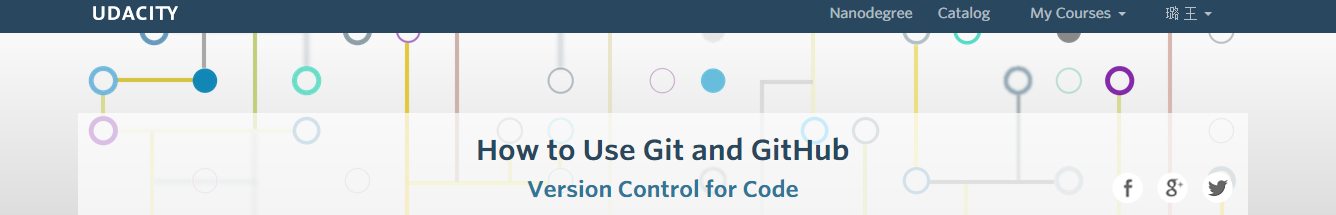


图2-10 Udacity的分享机制

**2.3.6 社会反馈**

描述：

社会反馈机制通常是用户接收来自他人的异步反馈的途径。它可以是由他人提交的“赞”，“踩”，或是评分，评论等。人们在提供社会反馈的同时也和他人产生联系，使MOOC成为一个巨大的学习共同体。收到他人反馈的用户会因此获得“自我认同感”，感受到自己在MOOC课堂中的重要性，增加与他人的互动，群体观念也会更加强烈，学习动机会被更强的激发。社会反馈分为结构化反馈和非结构化反馈两种。

结构化的社会反馈形式通常由计算机系统提供，通常只需要通过一次鼠标的点击来表现，例如点赞，踩，收藏，评分，标记帖子等。结构化的社会反馈不是一个分数，但是可计数的，并且会以数字方式呈献给用户，例如一个帖子获得多少赞，多少踩，被收藏了多少次，课程评分为多少等。

非结构化的社会反馈则通常是自由形态的，往往通过文本框等形式来实现。评论，回复等都属于此范畴。非结构化的社会反馈可能收到来自自身的社会反馈，例如讨论区主题帖中用户的评论可以被其他用户点赞。这也提供了一个反馈机制的循环，鼓励用户与他人分享，交流。

结构化和非结构化的社会反馈并不是相互排斥的，相反，他们可以被结合在一起使用。然而在用户提供结构化反馈的同时强制要求用户提供耗时较长的非结构化反馈可能会对用户造成困扰，使用户放弃对课程进行反馈。例如在用户对课程进行评分时强制要求用户发表评论，可能会使用户因为麻烦而放弃评分。社会反馈的设置可以是很精细的，例如当用户给予课程1星或2星的低平分时，系统会自动弹出对话框，请求用户反馈课程问题。

举例：

如图2-11，在可汗学院中，学生在观看完视频后可以以结构化的社会反馈的方式，以通过单选题选项提交自己对课程的感受，方便课程发布人评估本课时的学院满意度，及时对课程结构进行改进。图2-12为edX的反馈机制的应用，他采用结构化反馈和非结构化反馈结合的形式，在用户对课程内容，课程讲师，课程供应商进行星级评分的同时，被强制要求填写姓名，课程意见等信息。这种较为复杂的评价系统可能会使用户失去耐心，放弃评价。可以对此机制进行改进，将部分反馈由必填项改为选填项，增加用户自主性，也可以用物质激励来刺激用户，例如在完成课程评价后可以获得某个勋章，或增加积分等。图2-13为Udacity的反馈循环设计。在讨论区的帖子中，用户可以针对帖子主题发表评论，进行非结构化反馈，而每条用户的评论也可以被收藏，标记读过，进行结构化反馈，也可以被回复，形成非结构化反馈。

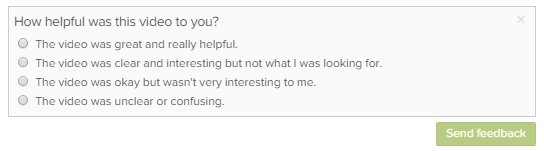


图2-11 可汗学院的结构化反馈

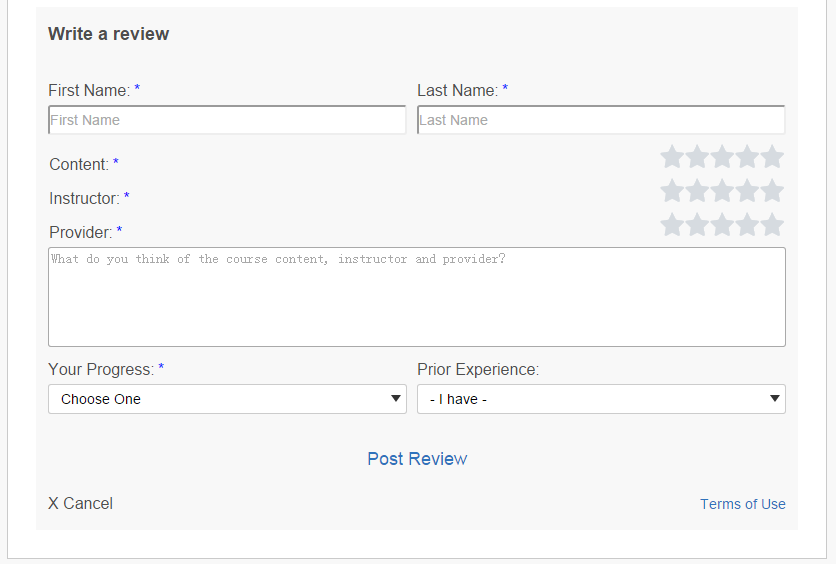


图2-12 edX的反馈机制

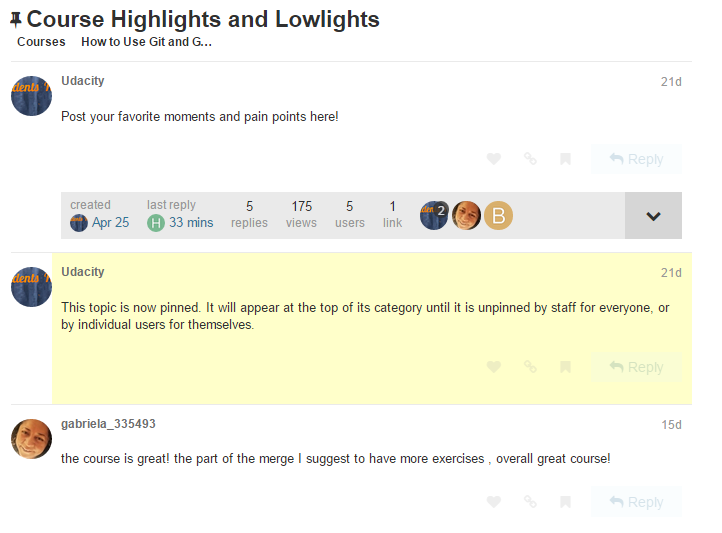


图2-13 Udacity的反馈循环

实现：

反馈机制中，统计帖子被赞，被踩数的结构化反馈的代码如下。

|  |
| --- |
| function create\_chattheme( )  {  $sql = "CREATE TABLE IF NOT EXISTS `chat\_theme` (  ……,  `upvote` int(5) DEFAULT 0 NOT NULL,  `downvote` int(5) DEFAULT 0 NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci";  mysql\_query($sql);  }//建表存储主题列表，upvote和downvote表示该主题被赞和被踩的数量  function count( ){  //如果是点赞  if($\_POST[''submit'']==''upvote''){  $sql=''update chat\_theme set upvote+=1 where user=$username;  }  //如果是踩  else if($\_POST[''submit'']==''downvote''){  $sql=''update chat\_theme set downvote+=1 where  user=$username;  }  mysql\_query($sql);  } |

反馈机制中，进行帖子回复的非结构化反馈的代码如下。

|  |
| --- |
| function reply( ){  //向主题回复表中插入回复内容  if($\_POST[''submit'']==''reply''){  $sql = ''insert into chat\_detail values(' ','$theme\_uid', '$user\_name', $\_POST['' reply\_content ''], date())'';  $this->db->query($sql);  }  } |

**2.3.7 状态保持**

描述：

用户可以在任何时候退出系统，系统会自动保存用户此时的状态。和Microsoft Office软件在用户退出时弹出提示框询问用户是否需要保存相比，MOOC平台通常会自动保存用户的学习状态，包括视频观看进度，作业提交情况等。因此，用户因系统突然崩溃而退出时仍能成功保存此刻的学习状态，这会给予用户以安全感和自由感。

举例：

如图2-14为中国大学MOOC的状态保持，当用户重新回到课程学习时，会显示上次的学习进度，并提供继续学习按钮，点击后会跳转到相应进度的课程页面。。图2-15和图2-16为Coursera的状态保持机制，用户的课程面板中会列出所有课程的学习进度，以百分比方式体现，如图中的项目管理课程，选择继续学习后，会跳转到上次离开时的状态，即继续通过还未通过的测验。



图2-14 中国大学MOOC的状态保持



图2-15 Coursera的状态保持（1）

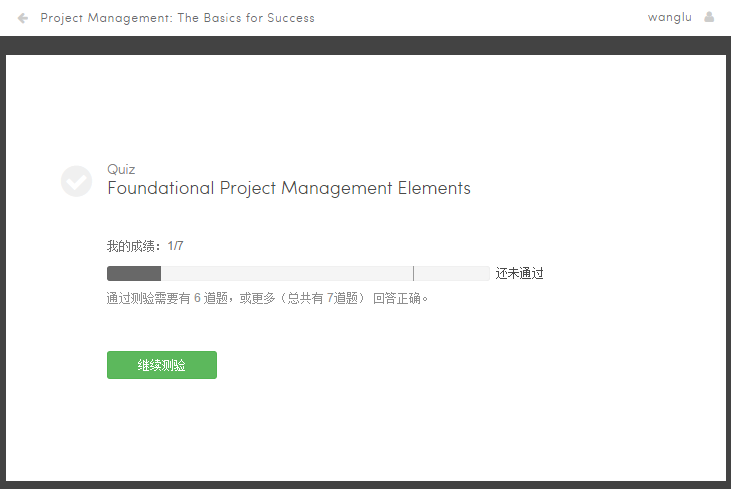


图2-16 Coursera的状态保持（2）

实现：

状态保持的实现如下。

|  |
| --- |
| //建立用户表时创建分数字段记录用户分数  //为user表添加状态字段  function createtable( ){  $sql = "create table user(  name varchar(100) not null,  ......,  state int(5) not null default 0 ) ";  mysql\_query($sql);  }  //创建课程表，主键为uid  function createtable2( ){  $sql = "create table course(  uid int(5) not null auto increment primary key,  name varchar(100) not null,  ......,  url varchar(100) not null,  state int(5) not null default 0 ) ";  mysql\_query($sql);  }  function savestate( ){  $sql="select \* from user where name=$username";  $user=mysql\_query($sql);  $sql="select \* from course where uid=$user->state";  $course\_now=mysql\_query($sql);  goto $course\_now->url;  } //找到当前用户状态信息对应的课时，获取其url并跳转 |

1. **iMOOC平台的需求分析**

通过前面对MOOC平台设计模式的研究，实现一个小型的MOOC系统——iMOOC平台，以下为iMOOC平台的需求分析。

**3.1 系统总体需求分析**

系统总体需求如图3-1所示。MOOC平台主要由课程详情模块，讨论区模块，作业提交模块，作业评分模块，考试模块，个人资料管理模块和留言模块构成，每个主体模块又由若干子模块组成。

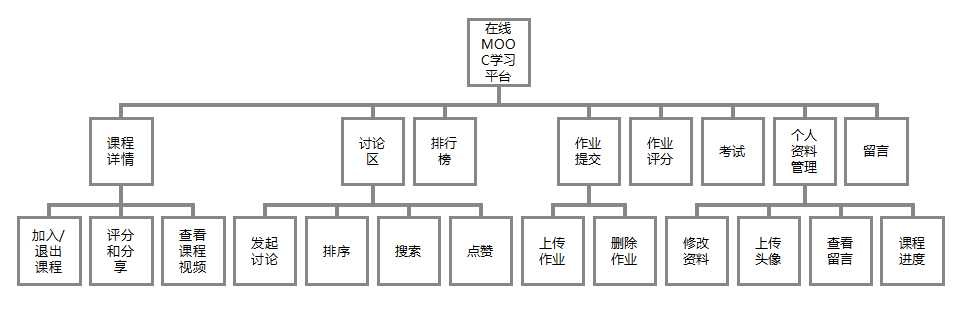


图3-1 系统总体功能模块图

**3.2 课程详情模块需求分析**

**3.2.1 课程详情模块的功能**

本模块主要包含了课程的基本信息，在用户加入和退出课程时会有不同的显示，主要功能为参加和退出课程，课程评分，分享。在用户加入课程时有留言和观看课程视频的功能。

**3.2.2 用例图**

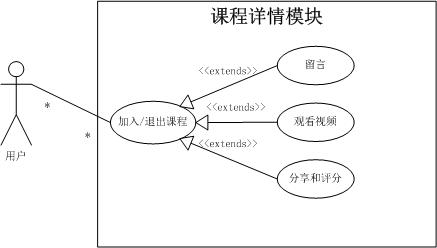


图3-2 课程详情模块用例图

**3.3 讨论区模块需求分析**

**3.3.1 讨论区模块的功能**

讨论区模块可以进行主题的查看；可以查看主题详情，且在不同板块的主题详情有不同的回复权限；可以进行留言；讨论区模块可以对主题进行排序，点赞和搜索；讨论区模块可以发起讨论。

**3.3.2 用例图**

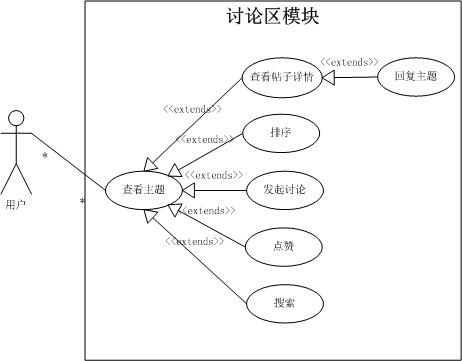


图3-3 讨论区模块用例图

**3.4 作业提交模块需求分析**

**3.4.1 作业提交模块的功能**

作业提交模块主要包含各周作业，在不同的时间段用户可以进行不同的操作。主要功能有作业的上传，查看自己上传的作业，与删除自己上传的作业。用户每周只能上传一份作业。

**3.4.2 用例图**

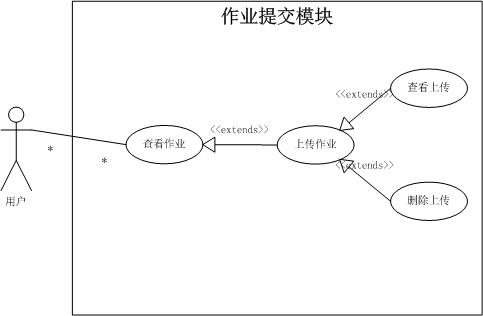


图3-4 作业提交模块用例图

**3.5 考试模块需求分析**

**3.5.1 考试模块的功能**

用户在考试模块仅有一次参加考试的机会。在选择进入考试后开始计时，在时间到时自动提交用户答案；在答案被提交后，由系统自动计算分数。

**3.5.2 用例图**

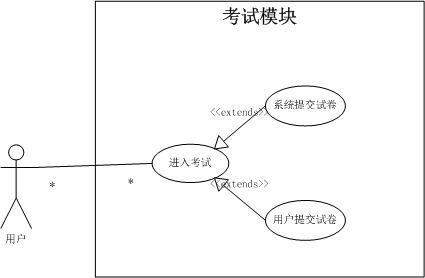


图3-5 考试模块用例图

**3.6 个人资料管理模块需求分析**

**3.6.1 个人资料管理模块的功能**

个人资料管理模块可以修改用户的基本个人资料；也可以上传用户头像；用户可以查看自己课程进度；用户可以设置自己的课程目标，并查看自己的达标情况；用户可以查看他人的留言。

**3.6.2 用例图**

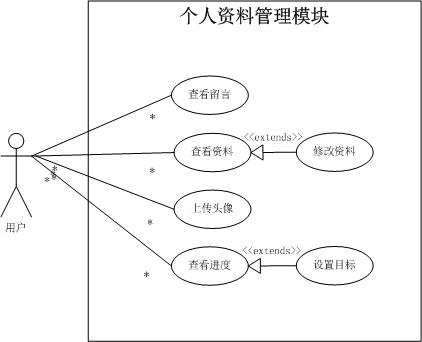


图3-6 个人资料管理模块用例图

**3.7 留言模块需求分析**

**3.7.1 留言模块的功能**

留言模块主要用来给相应用户留言，他和系统许多模块相关联，如排行榜模块，讨论区模块和课程详情模块出现的用户名和用户头像，通过点击可以进到相应的留言模块给其他用户留言，不同角色的用户可以查看的用户信息也不同。

**3.7.2 用例图**

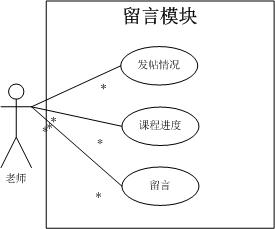


图3-7 留言模块老师用例图

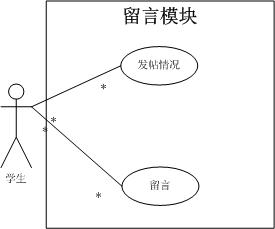


图3-8 留言模块学生用例图

**3.8 作业评分模块需求分析**

**3.8.1 作业评分模块的功能**

作业评分模块只有在用户角色为教师时才会出现，教师可以查看每周作业的学生提交情况，并对已经截止提交的作业进行评分。

**3.8.2 用例图**

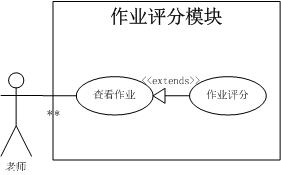


图3-9 作业评分模块学生用例图

**3.9 排行榜模块需求分析**

**3.9.1 排行榜模块的功能**

排行榜模块按用户总分，即考试加作业成绩排名。在排行榜中点击用户头像或用户名可以跳转到相应留言页。排行榜会显示用户当前排名。

**3.9.2 用例图**

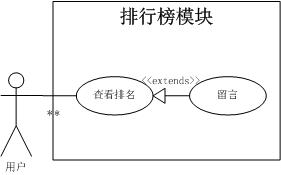


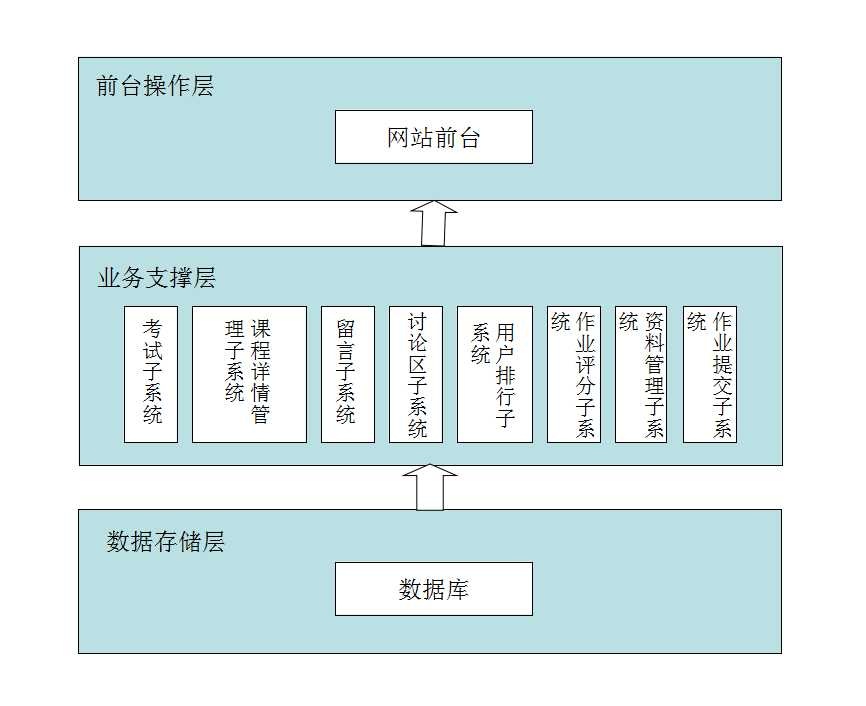
图3-10 排行榜模块学生用例图

1. **iMOOC平台的设计**

通过前面对MOOC平台设计模式的研究，实现一个小型的MOOC系统——iMOOC平台，以下为iMOOC平台的概要设计，模块详细设计和数据库设计。

**4.1 系统的概要设计**

iMOOC平台的系统架构图如图4-1。系统基于B/S架构进行Web平台的开发，使用基于 PHP MVC的CodeIgniter框架，将代码进行重构，分为表现层，应用层和持久层，增加系统的可维护性和可扩展性。



4-1 iMOOC平台系统架构图

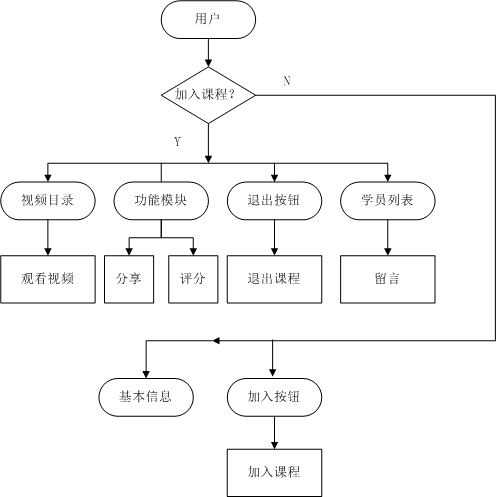
**4.2 系统模块的详细设计**

**4.2.1 课程详情模块的详细设计**

课程详情模块业务流程图如图4-2。在课程详情模块，用户可以加入和退出课程，页面因为用户的加入和退出呈现不同信息。

在用户退出课程时，会看到课程概述，课程评分要求，预备知识，授课大纲，授课老师等课程的基本信息。此时用户还可以分享课程，查看课程学习人数和课程评分，以及加入课程。

在用户加入课程时，会看到课程目录。点击每个课时对应的播放视频按钮会跳转到相应的课时视频播放界面，播放界面包含课时视频，分享按钮，以及该周的课程讨论区；此外，用户还可以进行课程分享；且除了查看课程学习人数外，用户还可以查看课程学员列表，并通过点击进入相应学员的留言页面进行留言；除了查看课程评分外，用户可以进行课程评分，且只能评分一次；用户可以退选课程。



4-2 课程详情模块业务流程图

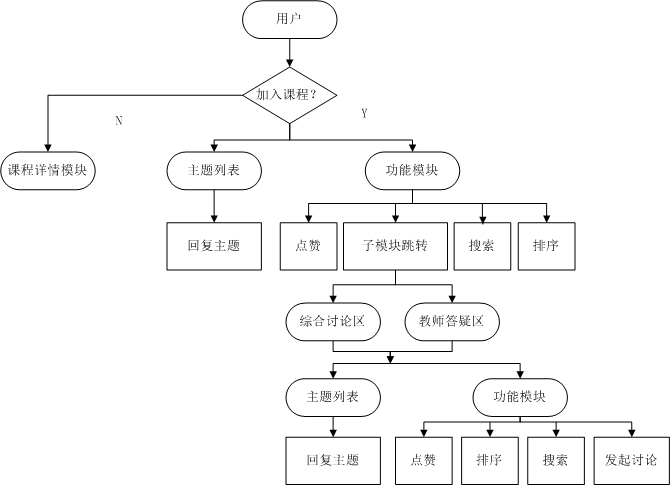
**4.2.2 讨论区模块的详细设计**

讨论区模块的业务流程图如图4-3。

讨论区模块的首页汇集了综合讨论区和教师答疑区的内容。在讨论区首页，可以分类查看主题帖，点赞主题帖，按特定顺序排序主题帖，进行搜索，和进入子讨论区。

在子讨论区，可以发起讨论，分类查看主题帖，点赞主题帖，对主题帖进行排序和搜索。在进入发起讨论页时，可以选择问题是针对某周还是常规问题，并键入帖子名称和帖子内容后发起讨论，此时用户会获得相应加分。

在查看主题帖时，可以对主题帖进行点赞，和发表回复，其中综合讨论区的帖子可以被所有人回复，教师答应区的帖子则只能被教师回复。此外，点击发帖人用户名可以进入到留言页面进行留言，留言后用户会获得加分。



4-3 讨论区模块的业务流程图

**4.2.3 排行榜模块的详细设计**

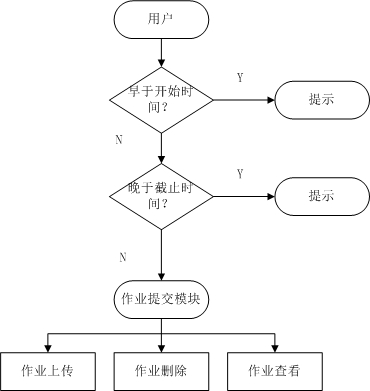
在排行榜模块中，可以查看用户排名信息，按用户课程总分进行排序，总分会存储在数据库中。显示信息包括用户头像，用户名，用户总分，用户排名。此外，在点击用户头像或用户名时，可以进入到相应的留言页面留言。

**4.2.4 作业提交模块的详细设计**

作业提交模块的业务流程图如图4-4。

作业提交模块列出了每周作业的开始时间和结束时间。点击查看按钮可以进入相应周的作业详情页面。

在周作业页面，如果此时时间在开始时间前，会出现“还未开始提交作业”提示，用户无法上传作业；如果此时时间在截止时间之后，会出现“作业提交已截止”提示，用户无法上传作业和修改已提交的作业，但可以查看已提交的作业。如果此时时间在开始时间和截止时间之间，用户可以选择作业上传，并查看已上传的作业，每周作业只允许学生提交一份作业，所以想要重新上传时，首先要删除已上传的作业。



4-4 作业提交模块的业务流程图

**4.2.5 作业评分模块的详细设计**

作业评分模块只有在用户角色为教师时才会出现在导航条上。和作业提交模块一样，它列出了每周作业的开始时间和结束时间。点击评分按钮可以进入相应周的作业页面。

如果此时时间在截止时间之前，则会出现“作业还未截止提交，无法评分”提示，但可以查看学生提交作业。如果作业已截止提交，则教师可以对学生作业进行评分，教师可以多次重复评分。

**4.2.6 考试模块的详细设计**

在考试模块，每个用户都仅有一次作答机会，。在用户选择进入考试后，考试页面计时器开始计时。考试题型为单选题和多选题如果用户点击提交按钮或计时器到达指定时间，用户的答案都会被提交到答案页面，系统自动比对用户提交答案和数据库中存储的正确答案并进行评分。

**4.2.7 个人资料管理模块的详细设计**

在个人资料管理模块，可以对用户基本资料进行修改，其中用户名和用户角色无法修改，密码和常用邮箱可以修改；用户可以上传头像，也可以保持系统默认头像；用户可以查看自己的课程进度，包括作业的提交情况和评分情况，考试的分数，讨论区发帖和回复的得分情况，以及课程总分；用户可以设置自己的课程目标，默认为60分，并通过进度条实时查看自己的完成度，完成度不同时进度条会呈现不同的颜色；用户可以查看他人给自己的留言。

**4.2.8 留言模块的详细设计**

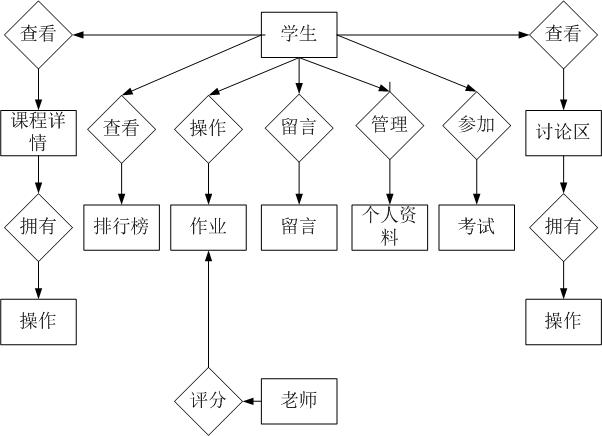
留言模块和排行榜模块，课程详情模块，讨论区模块等多个模块关联，在点击用户头像或用户名进入对应相应用户的留言页后，如果是学生用户可以看到该用户最近发表的主题帖，并对该用户进行留言。如果是教师用户，则除了学生用户可以看到的信息外，还可以看到对应用户的课程进度。

**4.3 系统的数据库设计**

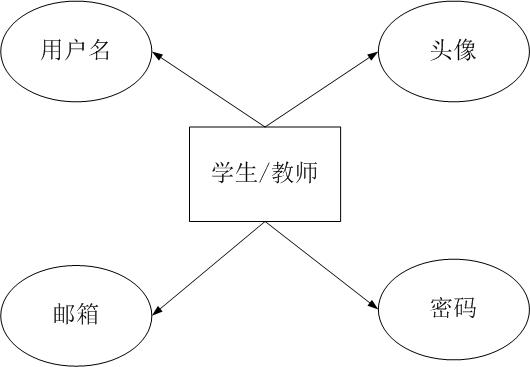
**4.3.1 数据库总体设计**

系统采用XAMPP建站软件集成包的MySQL数据库，采用utf-8编码，共有10个数据表。本章列举user,chat\_theme,chat\_detail,shomework四个数据表，其他数据表见附录A-F。

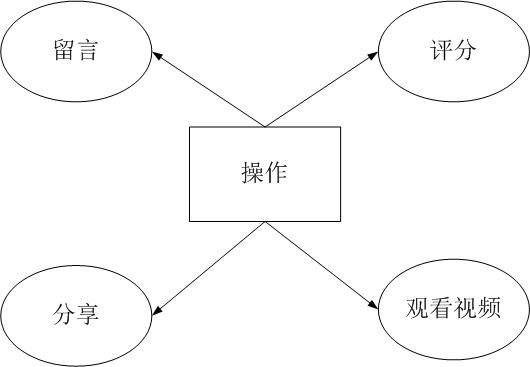
网站的总体E-R图如图4-5。其中教师可以完成学生的所有操作，除此之外，教师还可以进行作业评分，在图中不再赘述。实体和方法的属性如图4-6到4-10。



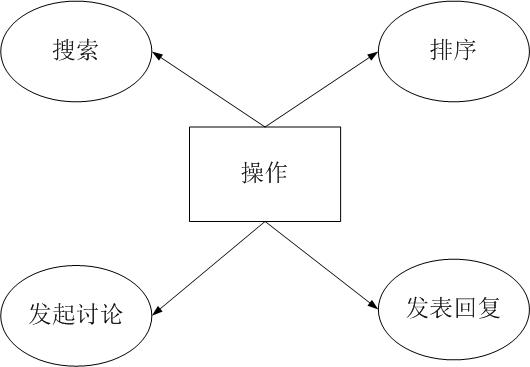
4-5 网站的总体E-R图



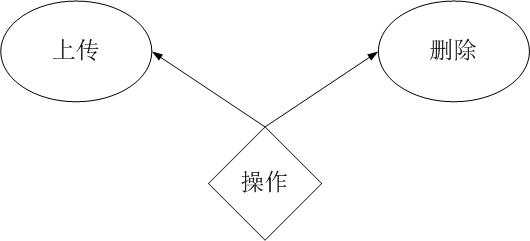
4-6 教师/学生实体的属性



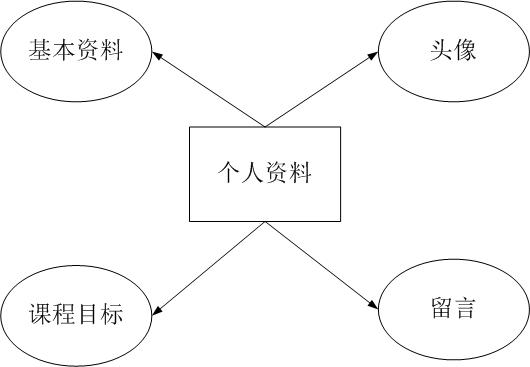
4-7 课程详情拥有的操作实体的属性



4-8 讨论区拥有的操作实体的属性



4-9 操作方法的属性



4-10 个人资料实体的属性

**4.3.2 user表的设计**

在用户进行注册时系统会将用户信息插入user表，当用户进行信息修改，图片上传，考试，上传作业等操作时系统都会对user表中的对应记录进行更新。User表的详细设计如表4-1。其中uid为主键；用户名name在注册时不允许重复；密码password用md5()加密存储在数据库中，email为用户注册时填写的邮箱；identity区分用户的角色是教师还是学生；enter表示用户是否加入课程，未加入时为0，加入时为1；doexam表示用户是否参加过考试，未参加过时为0，参加过时为1；examscore表示用户的考试分数；totalscore表示用户本门课程的总得分；touxiang存储用户头像的路径，默认头像路径为/mooc/touxiang/default.jpg；goal表示用户为自己设置的得分目标，默认为60分；chat\_theme表示用户发表主题帖的得分，每发一个得分0.5分；chat\_reply为用户发表回复的得分，每发一个得分0.2分。

表4-1 user表详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| uid | int(10) | 否 | 无 | 主键 |
| name | varchar(500) | 否 | 无 | 候选键 |
| password | varchar(500) | 否 | 无 |  |
| email | varchar(500) | 否 | 无 |  |
| identity | varchar(10) | 否 | 无 |  |
| enter | int(2) | 否 | 0 |  |
| doexam | int(2) | 否 | 0 |  |
| examscore | float(5,1) | 否 | 0.0 |  |
| totalscore | float(5,1) | 否 | 0.0 |  |
| touxiang | varchar(100) | 否 | /mooc/touxiang/default.jpg |  |
| goal | int(3) | 否 | 60 |  |
| chat\_theme | float(5,1) | 否 | 0.0 |  |
| chat\_reply | float(5,1) | 否 | 0.0 |  |

**4.3.3 chat\_theme表的设计**

chat\_theme表存储讨论区的帖子主题，每当用户发起新讨论时，会向表中插入一条新记录，主题名允许重复。详细设计如表4-2。其中uid为主键，和chat\_detail中的theme\_uid保持一致；user字段和user表中的name字段保持一致；theme字段表示帖子主题；content为发帖者键入的帖子原始内容；time表示发帖时间；chap为帖子对应章节，chap为0时表示是常规主题帖；identity为帖子对应的讨论区模块，可以是综合讨论区a或者教师答疑区t；love表示帖子被赞次数。

表4-2 chat\_theme表详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| uid | int(10) | 否 | 无 | 主键 |
| theme\_uid | int(5) | 否 | 无 | 外键 |
| theme | varchar(500) | 否 | 无 |  |
| content | text | 否 | 无 |  |
| time | datetime | 否 | 无 |  |
| chap | int(5) | 否 | 无 |  |
| identity | char(10) | 否 | 无 |  |
| love | int(10) | 否 | 无 |  |

**4.3.4 chat\_detail表的设计**

chat\_detail表存储讨论区的帖子回复，每当用户回复主题帖时，会向表中插入一条新记录。详细设计如表4-3。其中uid为主键；theme\_uid字段和chat\_theme表中的uid字段保持一致，为外键；user字段表示发表回复的用户名，和user表中的name字段保持一致；content为发帖者键入的回复内容；time表示发帖时间。

表4-3 chat\_detail表详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| uid | int(10) | 否 | 无 | 主键 |
| theme\_uid | int(10) | 否 | 无 | 外键 |
| user | varchar(50) | 否 | 无 |  |
| content | text | 否 | 无 |  |
| time | datetime | 否 | 无 |  |

**4.3.5 shomework表的设计**

shomework表存储了iMOOC平台的学生提交作业的信息。详细设计如表4-4。其中uid为主键；chap字段表示作业所属章节；fname为上传作业文件的名称；url为作业对应的url；sname为作业上传人的用户名，和user表中的name字段保持一致；score表示作业得分，默认为0。

表4-4 shomework表详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| uid | int(10) | 否 | 无 | 主键 |
| chap | int(5) | 否 | 无 |  |
| fname | varchar(500) | 否 | 无 |  |
| url | varchar(500) | 否 | 无 |  |
| sname | varchar(500) | 否 | 无 |  |
| score | int(2) | 否 | 0 |  |

1. **iMOOC平台的实现**

**5.1 网站开发环境**

开发语言：PHP

开发框架：CodeIgniter

数据库：MySQL

服务器：Apache

操作系统：Windows 7

建议浏览器：Firefox，Google Chrome

**5.2 注册和登录模块的实现**

如图5-2和图5-3，为iMOOC平台的登录和注册页面。在注册时用户名不允许重复，且密码采用md5()加密，其实现的程序流程图如图5-1。

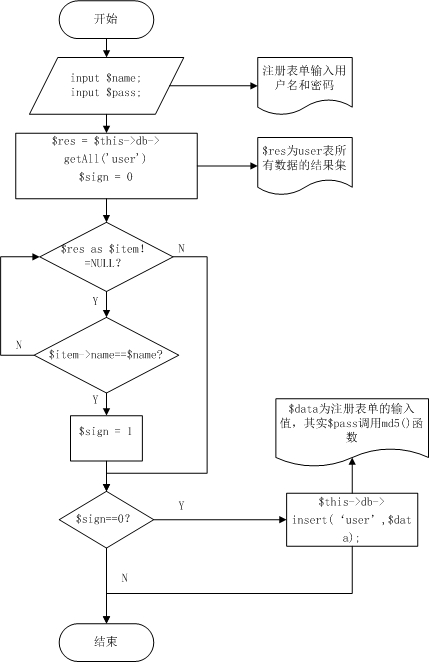


图5-1 注册的程序流程图



图5-2 登陆页面



图5-3 注册页面

**5.3 课程详情模块的实现**

如图5-6为未加入课程时的课程详情模块，图5-7为加入课程时的课程详情模块。实现用户加入和退出课程的操作的程序流程图如图5-4。

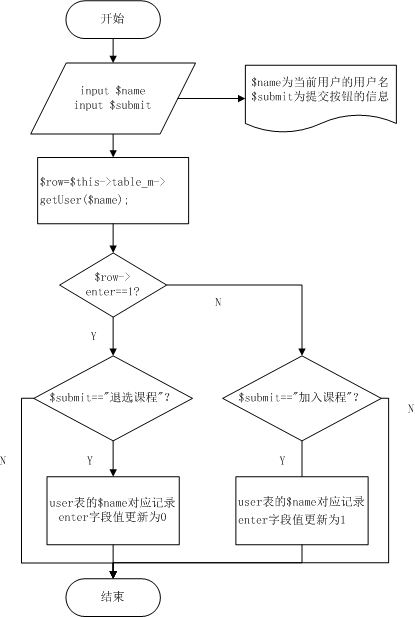


图5-4 加入/退出课程的程序流程图

当用户加入课程后，课程评分的实现如图5-5：

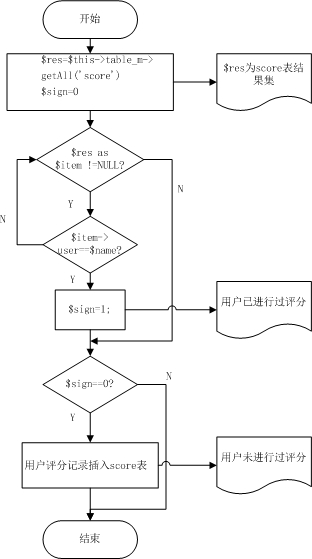


图5-5 评分的程序流程图

课程分享的实现使用JiaThis工具。

用户加入课程后，点击相应课时的播放视频按钮会课程视频播放页面，如图5-8。在视频播放页可以分享课时视频，可以通过点击链接跳转到上一课/下一课，右侧讨论区为该周所对应的的讨论区话题，可以按时间/按点赞数排序。



图5-6 未加入课程时的课程详情页



图5-7 加入课程时的课程详情页



图5-8 视频播放页

**5.4 讨论区模块的实现**

如图5-11为讨论区模块的讨论区首页。讨论区主题可以分周查看，右侧导航可以跳转到子版块综合讨论区/教师答疑区。在讨论区主题过多时，会产生分页效果，分页的配置代码如下。

|  |
| --- |
| //控制器中的分页配置及初始化  $config['base\_url'] =  site\_url( 'mooc\_c/chat' )."/".$i."/".$chap."/".$order."/".$search;  $config['total\_rows'] = $num;  $config['per\_page'] = $page\_size;  $config['uri\_segment'] = 7;  $config['first\_link'] = '首页';  $config['last\_link'] = '尾页';  $config['next\_link'] = '下一页';  $config['prev\_link'] = '上一页';  $config['num\_links'] = 2;  $config['use\_page\_numbers'] = TRUE;  $this->pagination->initialize( $config );  $data['links'] = $this->pagination->create\_links(); |

讨论区的搜索和排序功能均通过url传递的参数来实现，调用函数来取得讨论区主题帖数量和结果集，其实现代码如下。

|  |
| --- |
| //$i为讨论区对应板块，$chap为对应章节，$offset为偏移量，$page\_size为页面大小，$order为排序方式，$search为搜索条件  $num = $this->table\_m->getnum2( 'chat\_theme', $i, $chap, $search );  $res = $this->table\_m->page\_sel2( 'chat\_theme', $i, $offset, $page\_size, $chap, $order, $search); |

讨论区点赞的实现如图5-9。

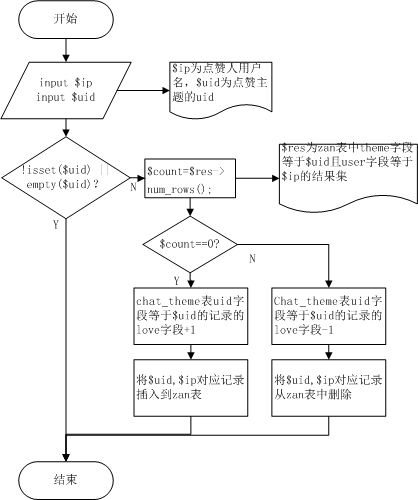


图5-9 讨论区点赞的程序流程图

综合讨论区和教师答疑区的实现情况和讨论区首页基本相同，在点击帖子主题进入帖子详情时，需要判断帖子所属板块，即chat\_theme中的identity字段，若是综合讨论区则所有人都可以回复，若是教师答疑区，则仅当用户的identity字段为教师时才可以回复。

在讨论区中，用户每发表一个主题帖，得0.5分，每发表一次回复，得0.2分，其实现的程序流程图如图5-10。

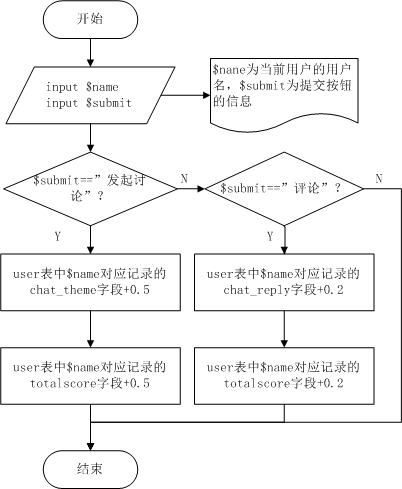


图5-10 讨论区加分的程序流程图



图5-11 讨论区模块首页

**5.5 排行榜模块的实现**

如图5-12为系统的排行榜模块，按分数从高到低排序，实现代码如下。

|  |
| --- |
| $res=$this->table\_m->table\_info('user','','totalscore desc');  $data['list']=$res; |



图5-12 排行榜模块

**5.6 作业提交模块的实现**

如图5-14为MOOC平台的作业提交模块，在点击查看按钮时，页面会跳转到如图5-15。当当前时间处于开始时间和结束时间之间时，会出现文件上传框允许上传作业。实现的程序流程图如图5-13。

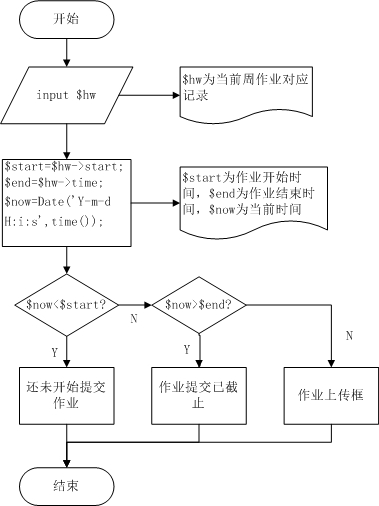


图5-13 作业上传的程序流程图

在作业还未截止提交时，用户可以删除已上传的作业。



图5-14 作业提交模块

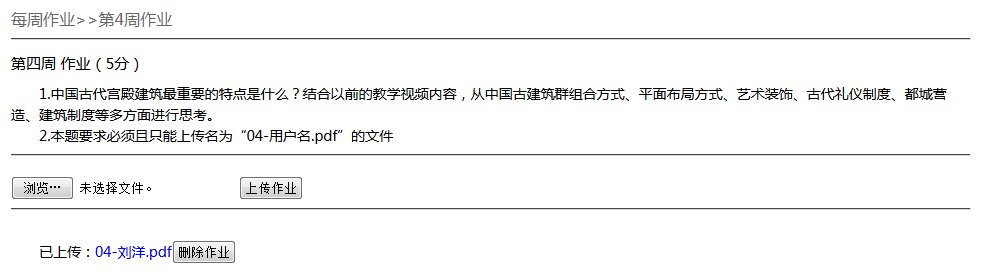


图5-15 每周的作业提交

**5.7 作业评分模块的实现**

作业评分模块的实现和作业提交模块类似，点击评分按钮后，跳转到如图5-16。仅当作业提交已截止时才能进行评分，在点击评分按钮时，homework表中会进行标记。评分的实现流程图如下图5-17。

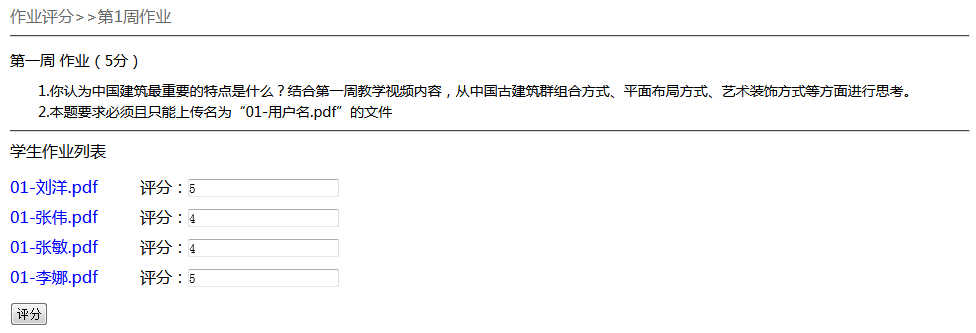


图5-16 每周的作业评分

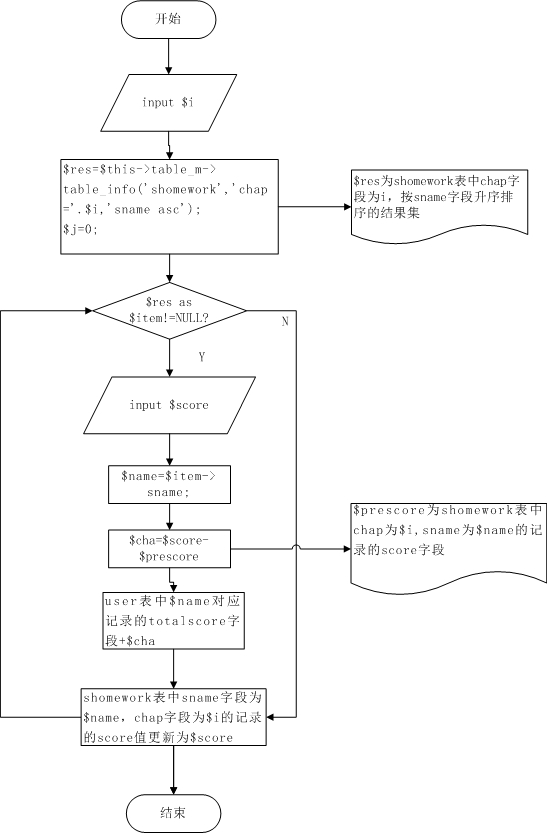


图5-17 作业评分的实现

**5.8考试模块的实现**

用户进入考试后界面如图5-18。在考试模块中，学生仅能参加一次考试，通过控制user表中的doexam字段来实现，在未参加考试时doexam的值为0，参加后值为1。考试题型为单选题和多选题，进入时计时器会开始计时，时间30分钟，超过时间系统会自动提交试卷。在试卷被提交后，进行用户提交答案和标准答案的比对，即exam表的ans字段，比对结束后计算用户考试分数，并更新user表的examscore字段和totalscore字段。其实现代码如下。

|  |
| --- |
| $this->table\_m->update('user','examscore='.$count,"name='$name'");  $this->table\_m->update('user','totalscore=totalscore+'.$count,"name='$name'"); |



图5-18 用户考试界面

**5.9个人资料管理模块的实现**

在个人资料管理模块中，点击“修改资料”按钮后，可以修改个人基本资料，修改完毕时候点击“提交修改”按钮，如图5-19。用户可以将系统默认头像变为自己喜欢的头像，如图5-20，点击浏览头像选择图片后，点击修改头像按钮，即可上传个性头像。

自己的课程进度也可以在该模块中查看，如图5-21。

用户可以设置自己的课程目标，默认为60分，完成度会以进度条的方式显示，如图5-22。其中完成度在0-20%时进度条完成部分呈灰色，在20%-40%时呈绿色，在40%-60时呈橘色，在60%-80%时呈红色，在80%-100%时呈蓝色。进度条实现代码如下。

|  |
| --- |
| $percent=round($user->totalscore/$user->goal \* 100 , 2);  <span class='red' style='width: ".$percent."%'>  <span>".$user->totalscore."/".$user->goal."</span></span> |



图5-19 修改基本资料



图5-20 上传头像



图5-21 课程进度

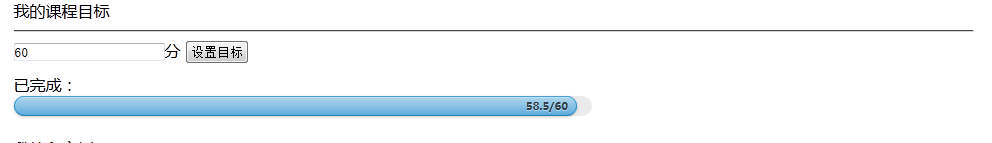


图5-22 课程目标

**5.10留言模块的实现**

在留言模块，不同身份的用户可以查看到不同的信息。如果是学生，可以查看对应用户发表的主题帖和进行留言如图5-23；如果是老师，则除此之外，还可以查看学生的学习进度，如图5-24。留言的实现如下。

|  |
| --- |
| function leave\_msg($to,$from){  $time=date('Y-m-d H:i:s',time());  $from=urldecode($from);  $to=urldecode($to);  $msg=$this->input->post('msg');  $arr=array('tosb'=>$to,  'fromsb'=>$from,  'con'=>$msg,  'time'=>$time  );  $this->db->insert('message',$arr);  } |

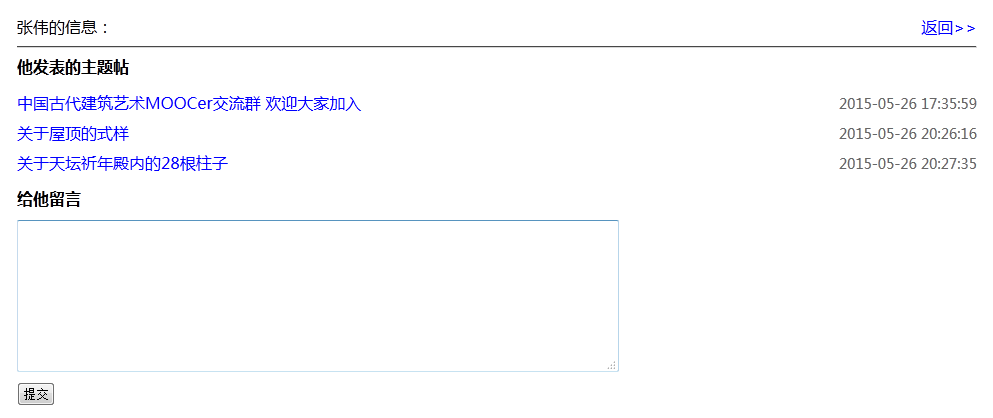


图5-23 学生查看留言面板



图5-24 老师查看留言面板

1. **iMOOC平台的测试**

在对iMOOC平台进行测试的过程中，共注册了5个用户，包括4个学生用户和一个教师用户，注册用户使用网站过程中，各项功能均正常运转，具体情况如下表6-1。

表6-1 iMOOC平台测试结果表

|  |  |  |
| --- | --- | --- |
| 模块 | 功能 | 运行情况 |
| 注册模块 | 用户注册（中文） | 正常 |
| 登录模块 | 用户登录（中文） | 正常 |
| 课程详情模块 | 加入和退出课程 | 正常 |
| 课程评分和分享 | 正常 |
| 查看其他用户信息 | 正常 |
| 进入课时页观看视频 | 正常 |
| 讨论区模块 | 发起讨论 | 正常 |
| 排序查看 | 正常 |
| 搜索（中文） | 正常 |
| 点赞和取消赞 | 正常 |
| 查看和回复帖子 | 正常 |
| 作业提交模块 | 提交作业 | 正常 |
| 删除作业 | 正常 |
| 作业评分模块 | 评分作业 | 正常 |
| 考试模块 | 参加考试 | 正常 |
| 排行榜模块 | 查看排行 | 正常 |
| 查看其他用户信息 | 正常 |
| 个人资料管理模块 | 修改资料 | 正常 |
| 上传头像 | 正常 |
| 查看进度和留言 | 正常 |
| 设置目标 | 正常 |
| 留言模块 | 用户留言 | 正常 |

综上所述，iMOOC平台通过网站测试。

1. **总结**

本课题研究了主流MOOC平台，如可汗学院，Udacity，Coursera，edX的设计，总结出一套现今MOOC平台通用的设计模式。且依赖于总结出的设计模式，实现一个简化版的MOOC平台-iMOOC，实现“中国古代建筑艺术”这门课程的在线学习。

由于MOOC属于近几年的新兴事物，在国内，并没有太多关于MOOC的资料可以借鉴，因此必须要学着去翻阅外文文献，这也提高了我的英语能力和研究能力。在进行设计模式研究的过程中，因为一开始对“设计模式”的理解不够深刻，软件工程的基础知识学的不够扎实，所以在研究上出现了偏差，也直接影响到了后面的编码工作，做了许多无用功，这也教育了我要在项目开始初期打好基础，磨刀不误砍柴工。网站的实现使用了PHP语言+MySQL数据库，并使用了CodeIgniter框架，虽然之前使用PHP开发过小型网站，但我却对框架这个概念没有太深的了解，在这次的开发经历后，我深深体会到框架的魅力，他会使我们的代码结构更加清晰，实现更加简单，代码的可读性和可扩展性也会变得更好，在以后的学习中，我也会对更多的框架进行深入研究，优化编码的过程。

虽然最初由于理解的偏差走了不少弯路，对毕设的工作量估计不足使得许多预想的功能没有实现，例如视频的进度控制，课程的导入等。但这次的毕设对我来说无疑是一次宝贵的实践经历，我也会认真的总结经验，吸取教训，争取在以后的项目开发中能做的更好。

**致谢**

首先，我要感谢我的毕设导师——赵丰年老师。尽管平日里工作繁忙，但在毕业设计的每个阶段，从最初的开题，文献翻译，资料查阅，到开始着手进行研究，到网站的设计和实现，赵老师都给予了我悉心的指导。在我的研究出现偏差时，是赵老师及时给予了我指导性的意见，帮助我理解课题，对论文和网站的编码进行修改。赵老师渊博的学识，严谨的治学态度和缜密的思维也将永远是我的榜样，是我在今后的日子里前进的方向。

此外，我要感谢我的母校——北京理工大学，母校在四年的时间里见证了我们的成长，我们的今天和母校提供给我们的优良的学习氛围密不可分。

最后，我要感谢在大学四年里教导过我的每一位老师，他们的谆谆教诲为我们奠定了知识的基石，使我们在这四年里渐渐充实，使我们未来能走的更远。

**参考文献**

[1]Luke Welling .PHP和MySQL Web开发（原书第4版）[M] .机械工业出版社,2009

[2]刘西杰,柳林.HTML CSS JavaScript网页制作从入门到精通[M]. 人民邮电出版社,2013

[3]王飞飞,崔洋,贺亚茹.MySQL数据库应用从入门到精通（第2版）[M].中国铁道出版社,2014

[4]李曼丽,张羽,叶赋桂.解码MOOC：大规模在线开放课程的教育学考察[M].清华大学出版社,2013

[5]Christopher Francis Lewis .MOTIVATIONAL DESIGN PATTERNS[D].UNIVERSITY OF CALIFORNIA SANTA CRUZ,2013

[6]Mary Treseler .Gamification by Design[M].O’Reilly Media,2011

[7]Detrick DeBurr.Build Gamified Websites with PHP and jQuery[M].Packt Publishing Ltd,2013

[8] Aaron Saray .PHP设计模式[M].清华大学出版社,2010

[9]程杰.大话设计模式[M].清华大学出版社,2007

[10]胡思康.软件工程基础[M].清华大学出版社,2012

[11]刘慧婷, 王珊珊.MOOC课程在线编辑系统的设计与实现[J].中国现代教育装备,2015

[12]王秀婷.基于edX的MOOC互动交流平台的设计与实现[D].哈尔滨工业大学,2014

[13]王颖,张金磊,张宝辉.大规模网络开放课程(MOOC)典型项目特征分析及启示[J]. 远程教育杂志.2013

[14]徐慧,姜建芳,乔丙立.基于ThinkPHP 的远程教学系统设计与实现[J]. 工业控制计算机,2014

[15]陈肖庚,王顶明.MOOC的发展历程与主要特征分析[J] .现代教育技术,2013

[16]李华,龚艺,纪娟.面向MOOC的学习管理系统框架设计[J].现代远程教育研究,2013

**附录**

**附录A video表的详细设计**

video表存储课程的视频信息。详细设计如表A-1。其中uid为主键；chap和class为视频对应的章节和课时；ccon为视频的标题；url为视频的存储路径；start表示该课时视频可以播放的开始时间。

表A-1 video表详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| uid | int(10) | 否 | 无 | 主键 |
| chap | int(5) | 否 | 无 |  |
| class | int(5) | 否 | 无 |  |
| ccon | varchar(500) | 否 | 无 |  |
| url | varchar(100) | 否 | 无 |  |
| start | datetime | 否 | 无 |  |

**附录B zan表的详细设计**

zan表存储了主题帖被赞的情况，每当帖子被赞或取消赞时，会向表中插入或删除记录。详细设计如表B-1。其中uid为主键；theme字段和chat\_theme表中的uid字段保持一致，为外键；user字段表示发表回复的用户名，和user表中的name字段保持一致。

表B-1 zan表详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| uid | int(10) | 否 | 无 | 主键 |
| theme | int(10) | 否 | 无 | 外键 |
| user | varchar(500) | 否 | 无 |  |

**附录C homework表的详细设计**

homework表存储了iMOOC平台的作业信息。详细设计如表C-1。其中uid为主键；chap字段表示作业所属章节；conn1和conn2均为作业要求；head表示作业标题；time表示作业截止时间，start表示作业开始时间；is\_marked表示该章作业是否已经被评分，已评分为1，未评分为0。

表C-1 homework表详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| uid | int(10) | 否 | 无 | 主键 |
| chap | int(5) | 否 | 无 |  |
| conn1 | text | 否 | 无 |  |
| conn2 | text | 否 | 无 |  |
| head | varchar(500) | 否 | 无 |  |
| time | datetime | 否 | 无 |  |
| start | datetime | 否 | 无 |  |
| is\_marked | int(2) | 否 | 0 |  |

**附录D exam表的详细设计**

exam表存储了iMOOC平台的试卷信息。详细设计如表D-1。其中uid为主键；qus字段表示问题；op1,op2,op3和op4为问题的四个选项；ans表示该道题的答案。

表D-1 exam表详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| uid | int(10) | 否 | 无 | 主键 |
| qus | varchar(500) | 否 | 无 |  |
| op1 | varchar(500) | 否 | 无 |  |
| op2 | varchar(500) | 否 | 无 |  |
| op3 | varchar(500) | 否 | 无 |  |
| op4 | varchar(500) | 否 | 无 |  |
| ans | varchar(50) | 否 | 无 |  |

**附录E message表的详细设计**

message表存储了iMOOC平台的留言信息。详细设计如表E-1。其中uid为主键；tosb表示留言是给谁，fromsb表示留言来自谁，均和user表中的name字段保持一致；con表示留言内容；time表示留言的时间。

表E-1 message表详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| uid | int(10) | 否 | 无 | 主键 |
| tosb | varchar(500) | 否 | 无 |  |
| fromsb | varchar(500) | 否 | 无 |  |
| con | text | 否 | 无 |  |
| time | datetime | 否 | 无 |  |

**附录F score表的详细设计**

score表存储了iMOOC平台的课程评分情况。详细设计如表F-1。其中uid为主键；score为用户对课程的评分；user表示评分人的用户名，和user表中的name字段保持一致。

表F-1 score表详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| uid | int(10) | 否 | 无 | 主键 |
| score | int(5) | 否 | 无 |  |
| user | varchar(500) | 否 | 无 |  |