

南通大學



本科毕业设计(论文)报告

题 目	基于 Django 和 Vue 的在线学习系统设计
--------	---------------------------

学生姓名: 张壮壮

专 业: 物联网工程

指导教师: 王进

完成日期: 2019 年 5 月 22 日

诚信承诺书

本人承诺：所呈交的毕业设计是本人在导师指导下进行的研究成果。除了文中特别加以标注和致谢的地方外，其中不包含其他人已发表或撰写过的研究成果。参与同一工作的其他同志对本研究所做的任何贡献均已在文中作了明确的说明并表示了谢意。

签 名：_____日 期： 2019.05.22

本论文使用授权说明

本人完全了解南通大学有关保留、使用学位论文的规定，即：学校有权保留论文及送交论文复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容。

（保密的论文在解密后应遵守此规定）

学生签名：_____指导教师签名：_____日期： 2019.05.22

摘 要

在线学习拥有着许多传统的课堂授课方式没有的优点，可以使得用户具有更加高效的学习效率；并且和传统的教育方式进行相结合，可以带来如下优势：用户重复学习性、不受时间地点的约束性等。另外本文研究系统会根据用户的评价和学习数据分析，对不同的用户做出智能课程的推荐，方便用户在海量的学习资料中筛选出需要的优质资源，从而获得良好的用户体验。

本文研究的系统是在 Django 与 Vue 技术的基础上设计实现的一套网络学习系统。采用前后端分离的架构模式，即前端将精力聚焦于页面的显示与用户的体验方面；而后端则将有限的精力集中的投入到不同的详细的业务逻辑处理中。利用 Django 搭建网站的后台管理与提供服务的相关 API,利用 Vue 技术进行前台页面的搭建。以此实现对资源的管理，并记录用户的操作行为，进行相关的总结形成邮件发送给用户。该学习网站包含用户端与后台管理两个子系统，具体分为六个功能模块，分别是用户、课程、机构、推荐、搜索、评论等模块。实现了快速开发到简单部署的整套流程助力教育资源的再平衡。帮助相关的教师更好的实现自身的教育理念。

本文通过测试系统，证明了该系统的具有极高的可用性。该系统通过对资源的管理整合为用户提供了便捷的学习环境与提供精确的月度总结报告。

关键词：在线学习；用户行为记录；信息管理

ABSTRACT

Online learning has advantages that are difficult to beat by traditional classroom teaching methods, enabling users to learn more efficiently. It can bring the following advantages: repeat learning and not limited by time and place, etc. In addition, the system in this paper will make intelligent recommendations to different users based on the evaluation and data analysis, so it is convenient for users to select high-quality resources from massive learning materials and obtain high-quality user experience.

The system studied in this paper is a set of network learning system designed, implemented and deployed on the basis of Django and Vue technology. And it adopts the architecture mode of front and back end separation, that is the front end will focus on the display of the page and the user experience; The backend focuses limited effort on the detailed business logic processing. Django is used to build the background management system of the website and relevant APIs of providing services, and Vue technology is used to build the front page. It enables the management of resources and the record of users, and generates emails sent to the user. The learning website system contains two subsystems: client and background management system, including six functional modules, including user module, course module, institution module, recommendation module, search module and comment module. It realizes the whole process from rapid development to simple deployment facilitates the rebalancing of educational resources. And it helps the teacher to realize their own educational ideas.

By testing the system, this paper proves that the system is highly available. The system provides users with convenient learning environment and accurate monthly summary report through the analysis of data.

Key words: online learning; User behavior record; Information management

目 录

摘 要.....	Error! Bookmark not defined.
ABSTRACT.....	Error! Bookmark not defined.
第一章 绪论	Error! Bookmark not defined.
1.1 研究背景	Error! Bookmark not defined.
1.2 研究现状	Error! Bookmark not defined.
1.3 主要研究内容	2
1.4 论文结构	Error! Bookmark not defined.
第二章 相关工作	4
2.1 相关技术介绍	4
2.2 相关技术介绍	Error! Bookmark not defined.
2.3 课题技术路线	6
2.4 本章小结	7
第三章 在线学习系统需求分析	8
3.1 系统总体需求	8
3.2 系统功能需求	8
3.3 系统非功能需求	Error! Bookmark not defined.
3.4 本章小结	Error! Bookmark not defined.
第四章 在线学习系统分析与结构设计	14
4.1 系统流程图	14
4.2 数据库的逻辑结构设计	Error! Bookmark not defined.
4.3 系统架构	21
4.4 本章小结	22
第五章 旅游网站系统详细设计与实现	23
5.1 后台管理系统的实现	23
5.2 用户模块功能	25
5.3 课程模块功能	27
5.4 机构模块功能	29
5.5 用户行为记录模块功能	29

5.6 系统测试.....	31
5.7 本章小结.....	33
第六章 结论与展望.....	34
6.1 本文主要成果.....	34
6.2 未来工作展望.....	34
参考文献.....	36
致 谢.....	37

第一章 绪论

1.1 研究背景

由于各个地方的经济水平存在较大差异，导致教育资源分配的平衡。随着移动互联网技术的迅速发展^[1]，电脑、手机已经成为中国网民最大的互联网终端^[2]，在线教育的兴起让人们看到教育资源的再平衡成为可能，根据新闻联播的报道，B 站更是成为了青年人学习的第一阵地。由此可见在线教育的具有的广阔市场。

因此，打造一个公益性的免费学习系统显得尤为重要，借此系统助力教育行业进步，为国家培养大量的人才，提高社会的整体受教育程度。网络学习一定程度上打破了我国师资力量分布的问题，使得来自不同地区的同学可以随时随地的通过网络获取自己想要的知识。在线直播、在线视频播放等技术的提升也是助力用户打破时间有空间的限制，随心所欲的徜徉在知识的长河中，做到真正的书海拾贝，努力的汲取知识，充实自己的学问或者是精神世界。它可以让学有有更多的学习机会，提高获取各种知识的体验，增加了学习的投入度与兴趣，更是破除了教育组织的界限与之前比较封闭的结构，使其朝着多样性的方向发展。

1.2 研究现状

教育部基础教育司 2018 年工作要点明确指出，努力让每个孩子都能享有公平而有质量的基础教育^[3]。因此有很多研究学者致力于研究在线教育的课题，并取得了非常骄人的成绩，在某些方面一定程度上的冲击了传统教育的格局，对实现教育资源的再平衡做出很多贡献。他们通过大量的调查与实地考察等方法，结合自身的技术力量，打造了不同种类的在线学习网站。如由很多大学加入并积极推动的大学慕课网，成为了很多大学生学习的圣地，用户可以在上面获取非常优秀的教育资源。尤其它的免费策略更是为其带来了大量的坚实用户，为打破了各个学校的教师资源的不均衡现象的作用是极大的。还有被央视点名表扬的已经成为年轻人学习的前沿阵地的 B 站，B 站通过对用户上传的资源进行审核的方式实现了共享，加上实时的弹幕也是留住了大量的年轻用户；云课堂也是通过和诸多学校或国际知名的组织如 TDE 等打造了一个全新的学习平台。

然而也有一部分专家学者认为在线教育不适合学生，这些专家认为学生没有甄别能力等、会占用大量的课余时间、无法为中学生提供一种建立正确世界观、人生观、价值观和意识形态的环境等原因。但是更多专家学者还是认为在线学习是适合中小学教育的。它可

以让学生更富有热情和锻炼学生的自主学习能力，并在提高社会的适应性方面具有相应的贡献。

1.3 主要研究内容

本文通过利用自己所掌握的 Django 与 Vue 等技术，采用前后端分离的设计架构，在一定的指导下，研究在线教育系统搭建和推广的技术支持的理论。进行搭建一个在线学习系统，致力于为学生提供网络学习服务。可以针对不同的学校情况提供不同的定制化方案，为全国尤其是经济欠发达地区的学子提供良好的在线学习环境。

而本文研究的系统采用了 Django 技术进行后台管理的搭建和提供前台需要的数据服务，利用 Vue 技术对前端界面进行构建。在 Django 社区有许多开源强大的第三方插件^[4]；可以通过及其简单的配置操作在不书写大量代码的情况下就可以实现一个很友好的管理系统；Django 自带 ORM 可以非常有效的与对象进行管理与关联；Django 具有强大的利用正则进行构建的 URL 模块，可以满足对 URL 的深度定制；而其自身的可插拔式的设计模式，可以更好地进行定制服务；针对前台页面的开发提供静态模板，即其独有的 MVT 设计模式，大大的简化了程序的开发周期；具有简洁高效的缓存支持机制，提高用户使用的流畅度体验；自带的路由映射功能，可以简单有效的实现复杂的路由配置；同时具有很强大的权限管理功能；Django 在安全性策略上也具有一定的长效机制，为开发者提供便利；最为重要的一点是其可插拔的设计理念为本文快速开发有定制提供了理论依据。

而前端使用的 Vue 的技术，在借鉴吸收了 AngularJS 的模板特点和数据绑定特点与 ReactJS 的组件化特点和虚拟 DOM 技术特点的基础上进行再创造而形成的。采用自底向上增量开发的设计方式^[5]。Vue 将自身的优势汇聚于页面显示与用户体验，通过从后台获取对应的数据从而渲染网站页面，其本质为两个方向的数据绑定。Vue.js 的响应式双向绑定数据^[6]，其中的中心思想为一切都是组件化编程，组件之间支持嵌套为本文研究的系统的快速开发部署提供了有力的理论依据。

本文研究的系统采用的当前软件行业的潜在标准前后端分离的系统架构模式，前端将有限的精力聚焦于页面的呈现与用户的体验方面，而相应的后端就更加集中的进行与之相关的细微的业务逻辑的处理，前后端各司其职。前后端分离具有前端资源静态化、后端数据 JSON 化、系统平台无关化、页面与数据的构架分离等特点，具有以下优势：前后端流量分离后流量大幅减少、前后台性能的提高、前后端系统实现与平台无关和技术无关、安全性方面的可以很大程度上针对性的优化、开发与构架的分离使得前后端分离的架构模

式得到业界的广泛认可。互联网的创新已经转变为一些快速且小而美的实现，因此为本文研究的搭建在线学习系统提供了坚实的技术支持。

1.4 论文结构

本文研究的系统为在线学习系统，分为管理员端的在线学习管理系统和用户端的在线学习使用系统。前两个章节对相关研究背景和采用的技术栈做了详细而深入的描述，深刻的对比了这个技术之间的优缺点，再结合本文研究的课题的实际相结合，最终确定本文研究系统的具体技术的采用；在之后的第三章，对课题的深入理解的基础上，继续研究相应需求，并且从功能性和非功能性两个角度进行分析潜在的需求和具体的环境之间的关系；在四章节中对如何架构整个网站进行相应的叙述，介绍了主要流程、数据库设计，并对不同的功能以及之间的依赖关系进行了详细的介绍。并在接下来的一章讲述了具体的功能的实现并进行测试工作，并展示了部分的核心源码来解决相应的问题；第六章则是对本网站的一些缺点和可以优化的地方进行叙述，并计划在未来的时间内进行修改，并对本课题的方向进行一定的展望。

第二章 相关工作

2.1 相关技术介绍

本文使用了 Django 技术作为后台开发语言，Django 框架是一个开源的 Web 框架^[7]，深入了解并利用 Django 语言的特性就可以利用少量的代码进行构建了一个完整的网站。其自身的设计是非常流行的 MVC 模式，具有常见的 MVC 具有的快速开发、运维简单、快速方便部署等显著的特点。并且在延续了之前优点后，Django 将视图分成 View 模块和 Template 模块两部分^[8]；且使用 ORM 技术映射为类简化了开发的难度；提供非常强大的、丰富的功能模块；具有非常容易入手学习，丰富的社区资源帮助学习事半功倍；开发的项目容易维护；可以优雅的数据库访问从而减小技术壁垒；具有优雅的 URL 映射：通过正则表达式来管理 URL 的用具有极大的灵活性等优点。加上拥有大量的免费开源的第三方插件可以使用使得 Django 成为迅速架构网站的最佳选择。

在此基础上引入了第三方插件 Xadmin 作为后台管理的技术通过简单的配置就可以非常便捷生成了一个后台管理系统，为开发带来了极大的便利。选择 Xadmin 的原因是：具有很高的兼容性；可以进行灵活的自定义布局；具有即时编辑能力；具有相关的信息菜单；方便的修改数据；具有灵活的数据视图；具有强大的数据筛选能力；支持书签操作等强大的插件机制。还引入了第三方的开源的在 Web 开发中不可缺少的 DjangoUeditor 富文本编辑器插件，为后台提供丰富的文本编辑操作；

在本文中利用 MySQL 技术对数据进行持久化操作，并记录数据表之间的关系。使用 Redis 将临时数据持久化到本地磁盘，NoSQL 系统因其高性能、高可扩展性的优势在大数据管理中得到广泛应用^[9]。Redis 是一款开源的、网络化的、基于内存的、可进行数据持久化的 Key-Value 存储系统^[10]，具有很好的扩展性。Redis 本身是运行在内存中的，数据可以通过持久化策略 ROB 或者 AOF 机制有效的将数据持久化到本地磁盘。在本文研究系统中使用 Redis 记录用户的登录的 session、用户操作行为和缓存页面数据等用来提升了用户使用的流畅度体验。Django 内部的中间件技术也为记录用户行为与判断登录状态做出巨大贡献。

前台使用了 Vue 技术对页面进行构建。Vue 是基于 MVVM 模式的响应式框架^[11]。该模型独立于特定平台，具有很强的灵活性、效率和可扩展性^[12]。响应式网页设计是移动互联网蓬勃发展的技术产物^[13]。并使用与之配套的 Vue-router 进行路由的管理，Vue - router 是 Vue 官方提供的路由^[14]，使得构建页面的应用变的很简单；还使用了基于 Promise 原理实

现的 axios 技术进行发送网络请求与后端进行数据的交互；在此使用了以模块化为核心的 webpack 技术对前端的代码进行构建并作为资源加载和打包的工具；之后还是用了 ESLint 对前端的代码进行行业标准上的代码规范；再之后就是使用了 NodeJS 与 NPM 包管理技术和一些 ECMAScript6 技术的语法等技术, JavaScript 是 Web 上一种功能强大的编程语言，用于开发交互式的 Web 页面^[15]Error! Reference source not found.。由于 Vue 的组件化为快速开发与部署提供了理论的可能性。

2.2 技术分析比较

现在一般的网站开发大多会选择很久之前流行的使用 SSH 框架进行，使用 HTML 或者是 AngularJS 或者是 ReactJS 作为前端语言来搭建一体化 Web 网站。SSH 框架的时代已经过去，Struts2 的自定义的标签具有很复杂的使用和创建过程且每一次的网络请求达到 Struts2 的后台都会创建一个新的 Action 进行服务，并因为要维护其调用作用域链而加大了系统内存资源的开销；Struts2 具有在现在看来很不合理的线程不安全设计模式；拥有繁复杂乱的配置文件使得代码之间的耦合度大大的加大并降低了代码的重用性；而 Hibernate 则是学习成本太大难以真正的做到优化相关的性能问题且使用不当会导致资源的大量消耗；不适用与快速开发搭建部署的小型网站并难以进行分离架构。而前端使用的技术，如 JSP 部分的代码会加大服务器的压力、拖慢开发的进度、且过分依赖 java 的开发环境、代码不具有可重用性；AngularJS 本身的学习成本很高且由于其对数据做了脏检查而导致性能方面比之 Vue 也略有不足；ReactJS 依赖于 Virtual DOM 性能方面比之 Vue 略有不如。Vue 是在借鉴了 AngularJS 的模板和数据绑定技术功能和 ReactJS 的组件化和虚拟 DOM 技术的基础上进行再创造而形成的，比它们有更为显著的优点。

而且以往的网站多事没有进行前后端分离的一体化 Web 网站，在搭建、性能、维护上都不如前后端分离的优势明显。前后端框架可以同时由不同的人进行搭建^[16]。且已经成为互联网开发行业的业界标准，通过 web 服务器和应用服务器的方式对前后端进行分别部署，有效的进行解耦代码的耦合度。

前后端不进行分离的架构以 JSP 为例会导致拖慢服务器运行速度：JSP 本身在第一次执行时需要将其翻译成 Servlet 导致的运行速度下降降低用户的使用体验，而且会携带大量的非核心数据的 HTML 和 CSS 的代码，这些本不是处理业务逻辑的数据却占据了大量的内存开销及网络传输时间导致页面响应缓慢；还有就是由于 JSP 中的代码几乎不具有太大的可重用性的原因会导致开发速度的下降、人力物力的消耗。因此面向服务的前后端分离的设计模式只需要在前台进行页面的渲染和用户使用体验的改善，而后端则是通过定制相

关的接口为前台提供需要用来填充页面的 JSON 数据，从而达到了后台的开发人员将有限的精力投入到具体的每一个业务逻辑上并可以降低后台的服务器压力。这样做的好处还有就是可以使用各自熟悉的语言对系统进行快速开发做到真正意义上的前后端系统的技术无关化且具有强大的解耦性和代码复用性；还有就是它可以提高用户的使用体验，前后端数据的分离可以将之前一个服务器的压力均衡的分摊到两台或多台服务器上做到真正意义上的负载均衡和提高服务器的利用率并降低后端服务器流量；并且在安全性能等方面都做到真正的针对性处理等。

因此，本文采用以快速开发著称 Django 为基础的后台搭建语言，使用到的第三方插件进行配置生成管理后台和提高富文本编辑器的功能；MySQL 数据库作为一款开源的关系型数据库系统,具备体积小、效率高、简单易用等特点^[17]，使用 MySQL 作为数据进行数据的持久化；使用内存数据库(Redis)的基本概念和运行模式^[18]，Redis 是一种广泛使用的非关系型内存数据库系统^[19]。前台使用 Vue 技术构建页面结合 Vue-router 的路由器对相关的路由进行管理，并使用 Vuex 和 axios 对资源的状态和发送 http 通信并结合 webpack 对前端项目进行资源加载和打包；使用 ES6 作为 JS 的语法和 NodeJS 技术进行前后端完全分离的网站的搭建与测试。

2.3 课题技术路线

本文采用 Vue 和 Django 分别为前后端的开发语言，对前端和后端进行分别开发。具体表现为：使用 Django 的第三方插件进行相关的服务实现；以 Python 实现的并以快速开发著称的 Django 框架对后端 API 进行设计与实现，为前端提供相关的数据接口；使用免费开源的 NoSQL 数据库 Redis 对临时数据进行缓存达到优化用户的使用体验的目的；并使用 Django 的中间件技术记录用户的行为操作，如登录时长，页面的停留时间等信息；前端使用 Vue 作为基础进行页面的构建，绘制不同的组件并进行重用；使用对前端页面之间的 URL 进行路由管理，使用 Vue-route、Vuex 和 webpack 等进行路由、资源状态和打包管理；利用 axios 技术实现前后端数据的交互；和一些相关的技术，如 ES6、NodeJS 等。

2.4 本章小结

本章重点介绍了一下，选择这个课题所使用的到的一些技术，如 Django、Vue 等知名框架进行简单的介绍。在分析不同技术的特点与本文所研究对象之间的关系之后，通过对比可选技术的优缺点和他们自身的设计模式背景下，最终对本文选取的 Django 后端框架与 Vue 前端框架的原因给出了一定的解释。在系统架构方面，在阐述了前后端分离的历程，

分析前后端分离带来的优点与缺点之后，给出写本文所采用的前后端分离架构的技术理论支持。因此本文采用前端页面与后台的数据分离的架构模式，具体表现为后端的 Django 技术与前端的 Vue 技术。

第三章 在线学习系统需求分析

3.1 系统总体需求

本文研究的系统是为了解决学生的公益性免费在线学习系统的问题，而提出可以快速设计开发实现并部署的小型在线学习网站，针对不同的地区和学校的不同情况做出差异化定制化的系统实现。在万众创新的新时代中，互联网创新已经转变为一些快速的小而美的实现，因此本文研究系统也算的上是一次平凡的升华与创新。

在国家强大的基处建设中，互联网高度发达的情况下，网络已经成功的进入各家各户，人人都可以通过网络与外部世界连接。且人们对学习投入极大的热情和互联网产品的普及，所以就造就了在线教育市场规模空前巨大的现象。只需要在一定的技术支持的情况下在线视频播放变的非常容易。因此，实现优质教育资源的分享、搭建公益性的在线学习系统就显得很重要，如此才能真的实现教育资源的均衡，使得万千学子拥有平等享受教育资源的权利，可以在培养学生用户自觉能动性上做出巨大的贡献，并在一定程度上解决师资分配不均衡的问题。

3.2 系统功能需求

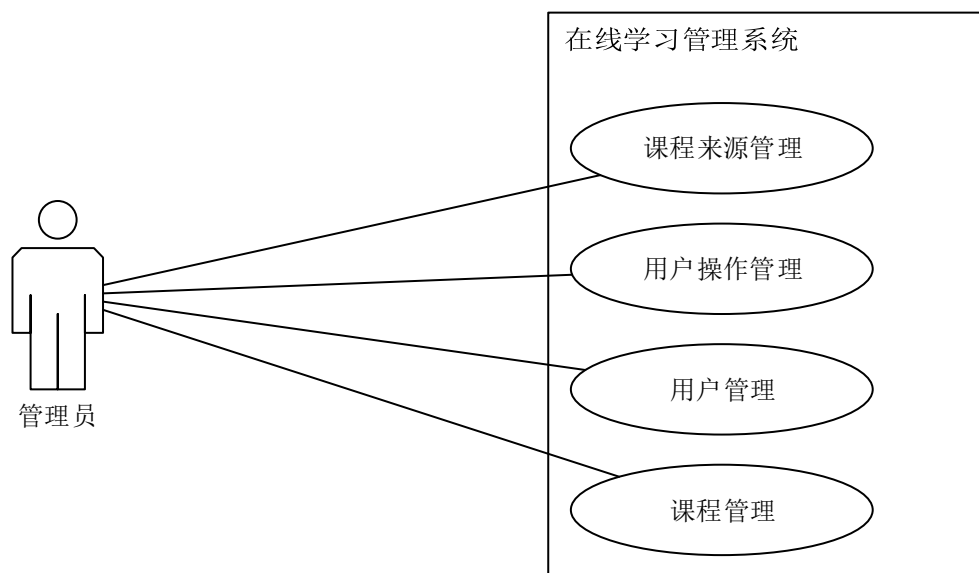


图 3.1 管理系统用例图

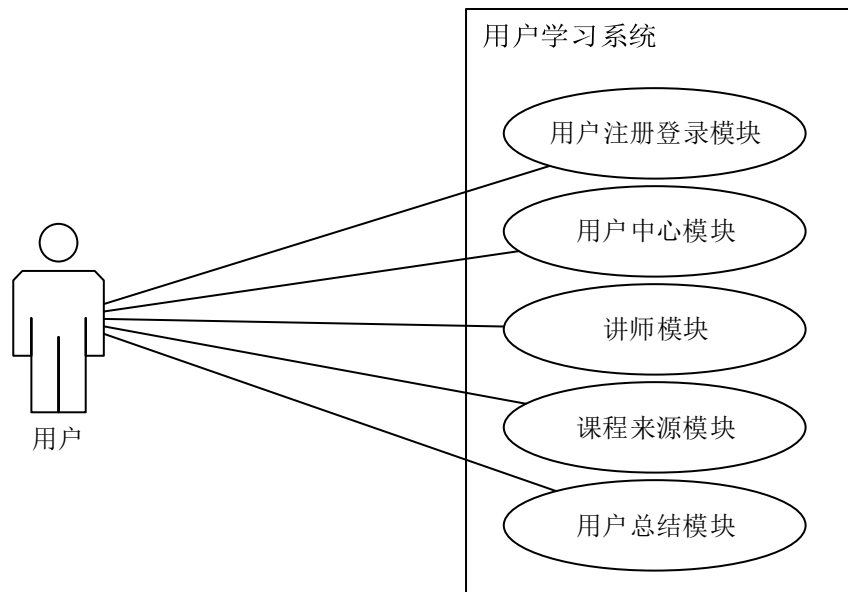


图 3.2 在线学习系统用例图

本文研究的系统包含后台管理系统和在线学习系统两个子系统，如图 3.1 和图 3.2 所示。管理员端的系统可以满足管理员对网站的整体掌握和管理使用，并对用户有相应的管理；用户端的在线学习系统可以为用户提供优质的学习服务，使得用户有良好的使用体验。

管理员有一定的课程来源于和教师信息管理需求，所以本文研究系统包含有课程来源与教师信息管理模块，可以方便管理员在后台对城市信息、课程机构信息、讲师信息进行相关的过滤、搜索、增加、修改、删除、查询等相关的操作。具体用例图如图 3.3：

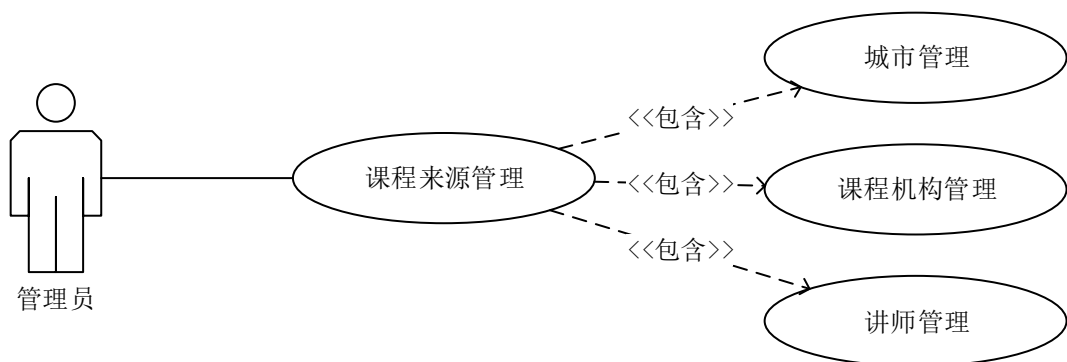


图 3.3 课程管理用例图

管理员可以通过平台对轮播图和用户信息进行管理，可以对数据进行导出为 Excel 或 JSON 等格式的文件和简单的增删改查操作。用例图如图 3.4 所示：

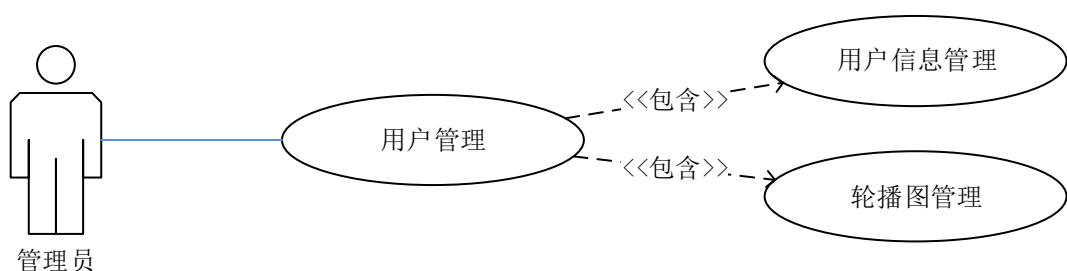


图 3.4 用户管理用例图

管理员对在后台管理用户的操作有一定的需求，本文研究系统具有相应的管理模块，可以对用户的操作行为，如收藏教师、收藏机构、评论课程、学习课程等方面都有一定的管理。管理员可以对查看不同课程、机构的收藏情况、不同用户的收藏情况、根据时间过滤收藏情况、导出收藏相关的管理。管理员在了解这些信息的基础上对网站的内容进行相关的优化，达到为用户提供更好地更加优质的服务，使得用户有更好的使用体验。具体用例图如图 3.5 所示：

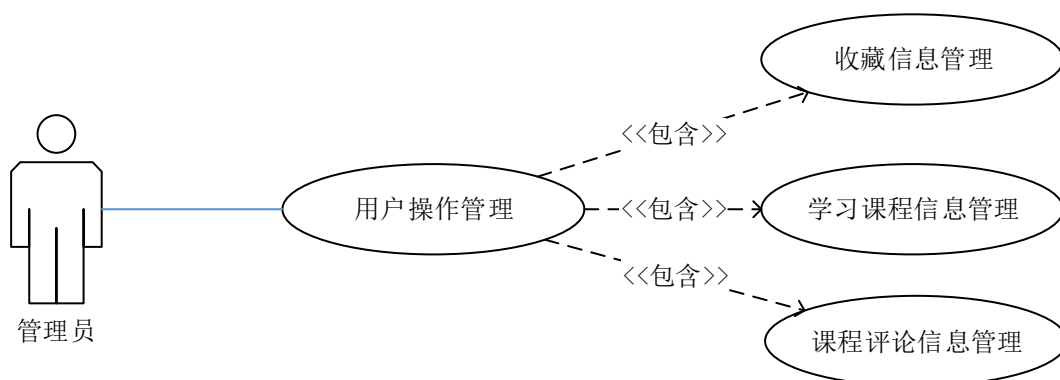


图 3.5 用户操作管理用例图

管理员需要可以上传课程信息，因此本文研究系统的后台管理系统拥有课程管理模块。不同的课程细分为不同的章节，可以对其进行管理的更加细化，尤其是课件资源信息更是优化的解决了课件资源难以找到或讲课件资源以链接形式发布在评论中的问题，使得用户可以更加轻松的得到相关的资源。具体的用例图如图 3.6 所示：

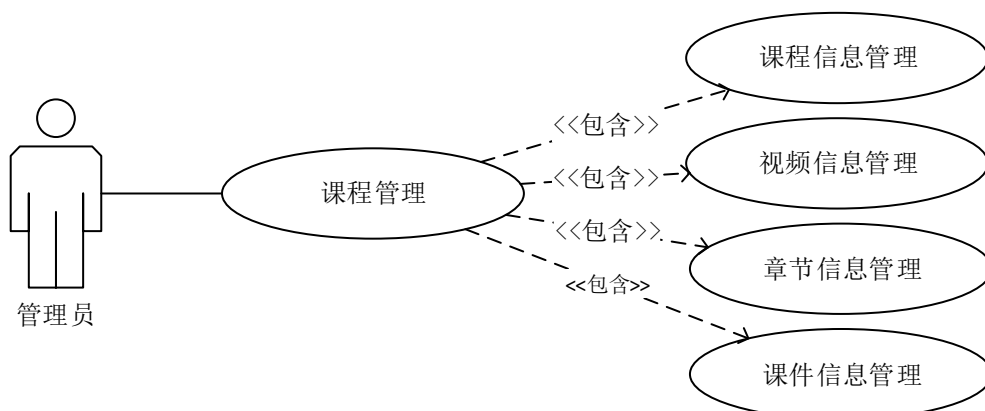


图 3.2.6 课程管理用例图

因为本文研究的是在线学习这样的—个长久性的服务，在用户方面要做到持久性的服务，因此获取的用户信息要有所验证，比如要验证这个邮箱确实属于该用户，方便以后的服务，如用户的月末总结邮件等。用户端的在线学习系统包含登录注册等模块。用户的登录状态会直接影响收藏课程、评论课程等相关的操作。具体的用例图如图 3.7 所示：

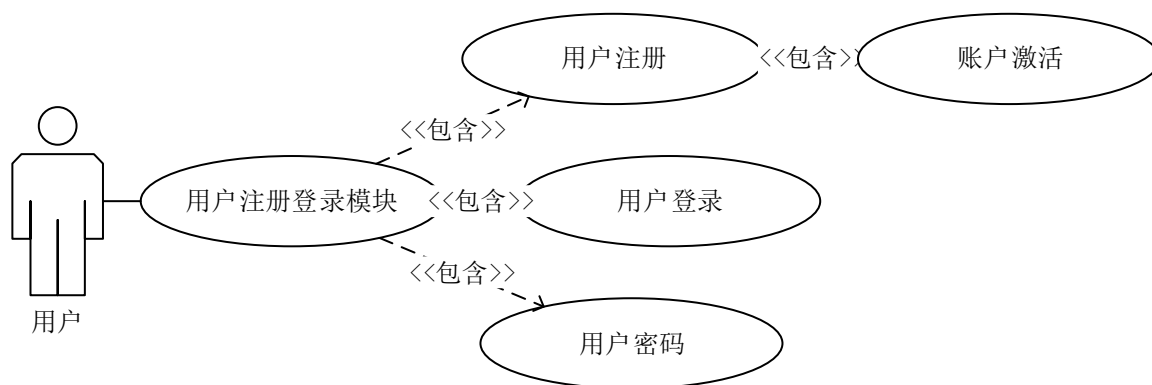


图 3.2.7 用户注册模块用例图

由于本文研究的系统具有长久性服务的特点，因此设计了用户中心模块，包含用户的个人信息，如性别、年龄、地址、手机号等选填内容，为以后的数据分析定制服务提供数据基础。用户的收藏信息，如收藏的课程、收藏的机构等信息，方便用户的学习与使用。用户模块还包含正在学习的课程信息，方便用户了解自己的学习情况。具体的用例图如图 3.8 所示：

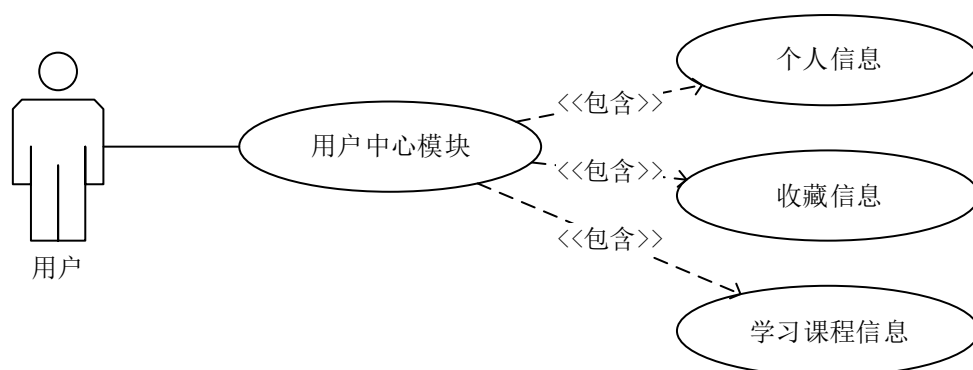


图 3.8 用户中心模块用例图

用户端在线学习系统还包含讲师模块，包含对讲师的列表的展示，与讲师排行榜的定制。方便用户具体的了解该讲师、收藏讲师、查阅该讲师的相关课程等信息。具体的用例图如图 3.9 所示：

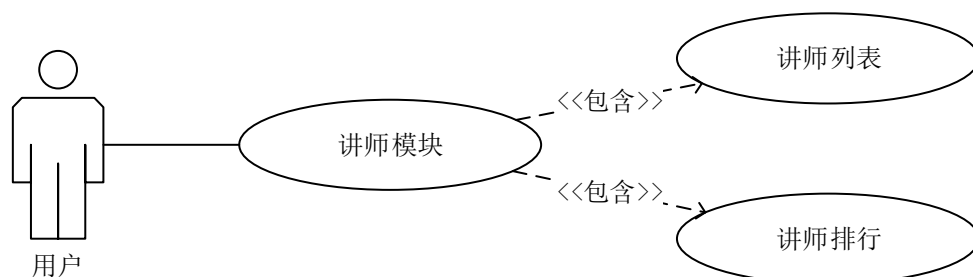


图 3.9 讲师模块用例图

用户端在线学习系统还包含课程来源模块，包含课程来源列表的展示和课程机构的排行榜展示。使用不同的过滤条件，如机构的类别、所在地、学习人数、课程数等条件进行

组合过滤。课程机构又包含机构的简介、机构的讲师、机构的课程等信息。具体的用例图如图 3.10 所示：

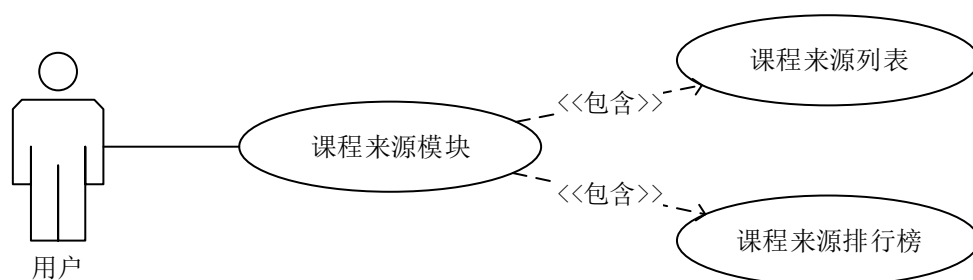


图 3.10 课程模块用例图

用户端在线学习系统还拥有用户行为模块，包含用户的登录的时长，学习的课程信息，在每个页面的停留时间、月末的用户学习情况总结。并分析这些数据做出相关的用户推荐和定制化服务、深度优化网站的性能等方面有很大的作用。用户总结模块如图 3.11 所示：

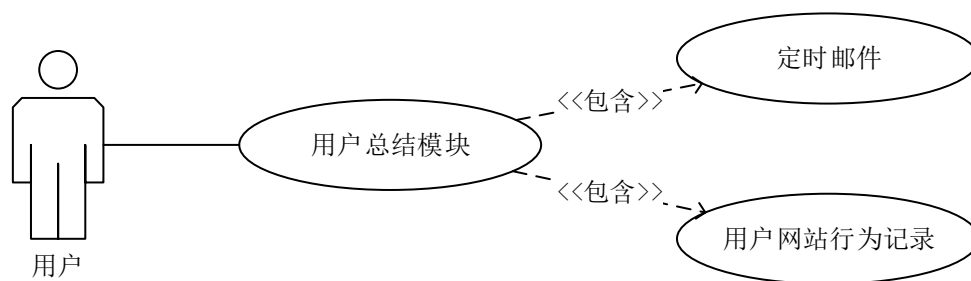


图 3.11 用户总结模块用例图

3.3 系统非功能需求

由于本文研究的系统需要对用户行为进行记录，因此会出现大量的数据和频繁的数据库操作等行为，性能上略有欠缺，目前页面一般的操作行为的响应时间为 5 秒内，比较好。由于存在大量的视频资源和课程信息资源，所以本文系统在服务器带宽与容量上有着一定的要求。系统由于是使用已经非常成熟的 Django 框架开发的，其自身具有跨站点脚本保护和跨站点请求伪造保护等安全措施，使得本文系统在安全性上具有极高的保证。在部署方面就显得极为容易，只需要将前后端分别部署到 web 资源服务器和应用服务器上即可。且本文系统所采用的框架本身特性，Django 框架的可插拔的设计理念和 Vue 框架的组件化编程思想决定了本文系统具有很大的可修改性、可用性和易用性，可以实现快速开发与部署。

本文所研究的系统是针对在线教育问题，实现线下的课堂教学与线上自主学习相结合的理念，为学生提供优质的在线学习服务。使得学生具有很好的学习体验，获取更多更优雅的知识。当然这会在法律规定的范围内使用，作为知识密集型的在线教育一定要遵从国家的知识产权保护法的相关规定，做到不侵权不偷窃别人的劳动成果。否则，人们无视法

律的尊严，肆意的剽窃偷取他人的劳动成果，就可能导致社会创新立的缺乏等问题。因此，在使用他人课程时一定要争取他人的同意，如此才能构建和谐美好的社会主义新风向。且本文研究系统应当为公益性的免费系统，想一定层度上破除师资力量分配的问题，不能使之成为谋取利润的工具。

3.4 本章小结

本章在分析了当前社会的在线学习人数的急剧增加和网络学习具有大量市场的背景下，分析线上学习与线下学习相结合的方式的优点的情况下，可以解决很多实际的问题，如课堂掌握不牢固等。在此基础上，本文基于如此需求，深入分析探究了技术的可行性方面之后，并结合传统课堂教学的特点，给出了如下的系统设计：系统分为后台管理系统和用户使用端；后台管理系统分为课程来源模块管理，对课程来源进行管理、用户模块管理，对当前用户进行管理，并配置首页的轮播图、课程模块管理，分别对课程、章节和章节下的视频或学习课件资源等进行管理、用户行为模块管理，对用户的收藏信息，用户学习信息，用户的评论信息进行管理；用户在线学习系统包含用户中心，可以进行注册登录修改密码等操作、讲师模块，对课程的讲师进行管理展示与介绍、课程模块、课程来源模块、推荐模块、用户行为模块、搜索模块和每月对用户行为进行总结。

第四章 在线学习系统分析与结构设计

4.1 系统流程图

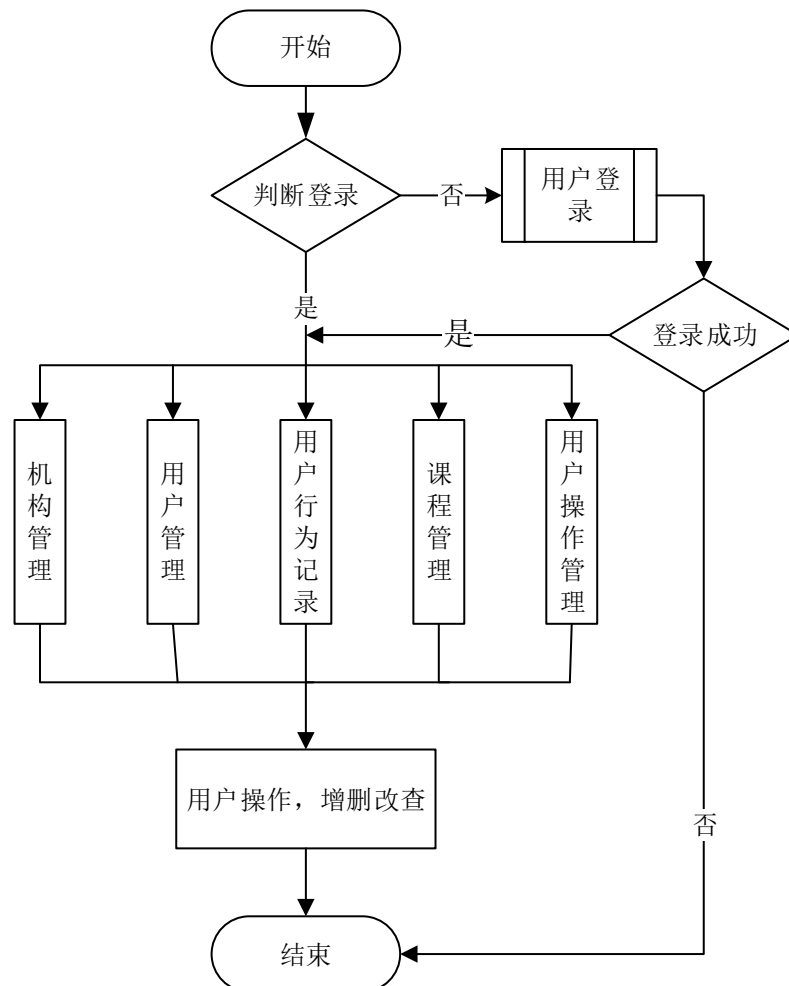


图 4.1 后台管理系统工作流程图

如图 4.1 所示为在线学习系统的后台管理系统，可以对机构信息、用户信息、用户行为、课程信息、用户操作等信息进行具体的操作并对数据进行多条件的过滤。只要是已经登录的管理员就可以对这些数据进行相关的管理，包含按照一定条件过滤数据、查看数据、排序、添加数据、删除数据、修改数据、导出数据为 Excel 或 JSON 等格式等。

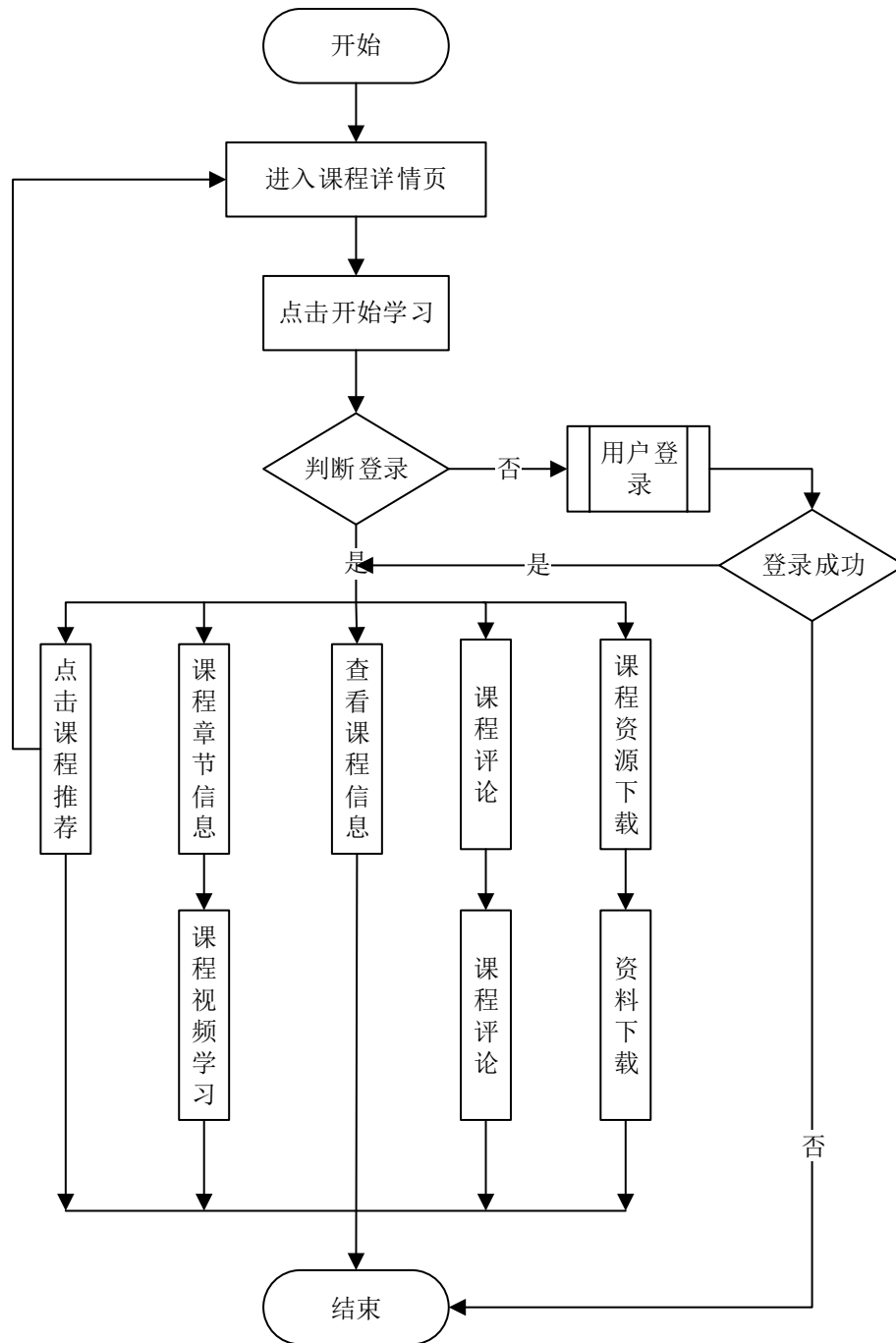


图 4.2 课程学习模块的流程图

如图 4.2 所示，为课程学习模块的流程图。在用户没有登录的状态下用户可以浏览课程的相关信息，但是却不能进行相关的课程学习与该课程的课件资源的下载。当用户点击学习该课程按钮时，在会自动的跳转到网站的登录页面并完成登录之后自动跳转到之前点击学习的课程的详细页面，不需要用户进行再次在网站中寻找该课程；登录之后的用户就可以对该课程进行学习。课程学习包含相关的课程推荐，该课程的相关信息，该课程的课件资源下载，课程章节视频学习与课程的评论等几个元素，用户可以对资源进行下载、对课程进行学习和对课程进行评论的操作。

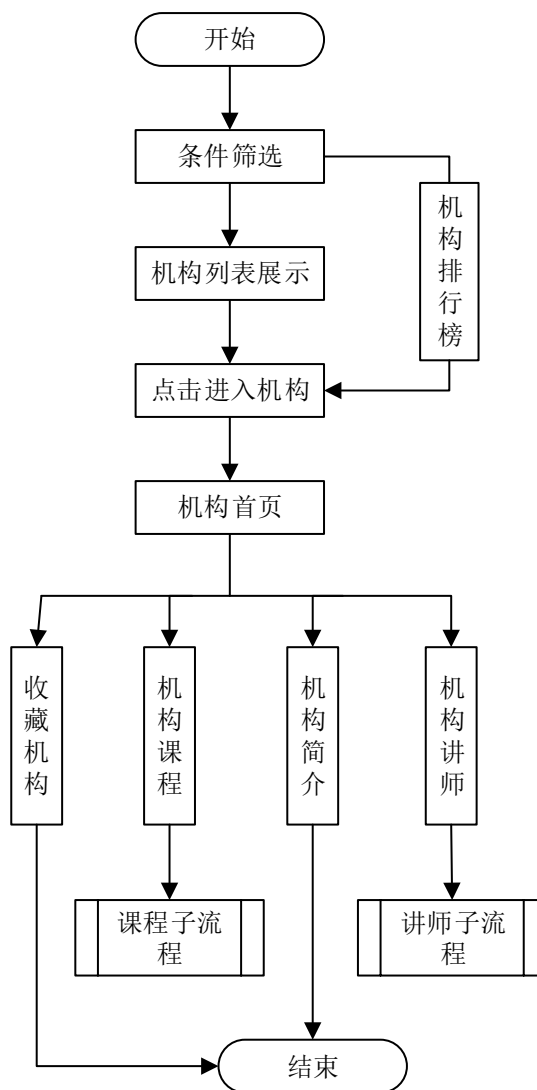


图 4.3 机构模块的流程图

如图 4.3 所示，展示了课程机构模块的操作流程图，用户通过该流程可以详细的了解到课程机构的相关操作。用户可以通过使用地区、类型、学习人数、课程数等条件对课程机构进行过滤查找，找到需要的机构，并且按照收藏数量展示了该条件下的机构排行榜。点击列表中的任何机构可以进入该机构的详情页，包含该机构下的所有教师、课程、简介等相关的信息，点击其中的课程或者讲师就会进入相应的课程或者讲师的流程。

4.2 数据库逻辑结构设计

数据库表设计的课程管理模块的实体图如图 4.4、图 4.5、图 4.6、图 4.7，分别用于记录课程信息、课程的章节信息、课程的课件信息、课程的章节视频信息；

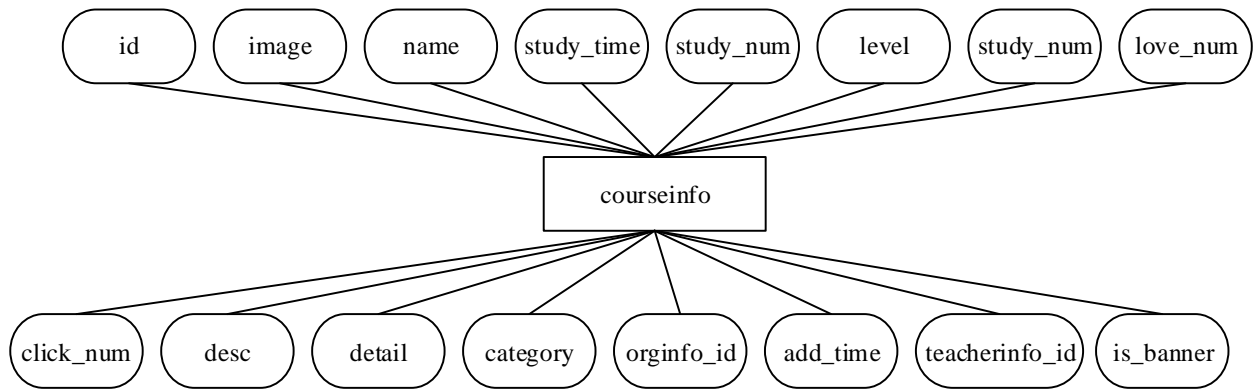


图 4.4 课程实体图

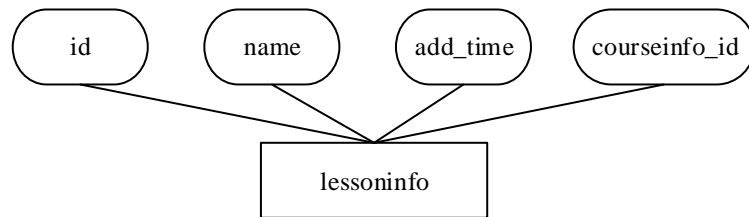


图 4.5 课程章节实体图

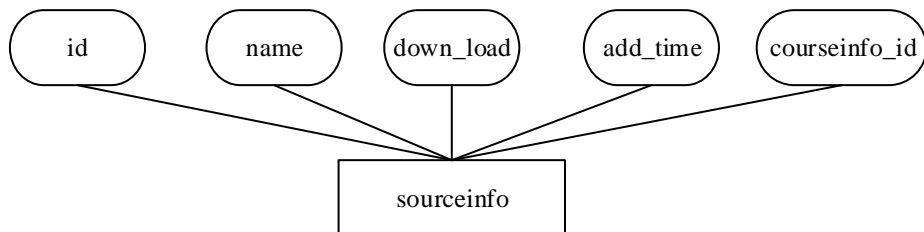


图 4.6 课程资源实体图

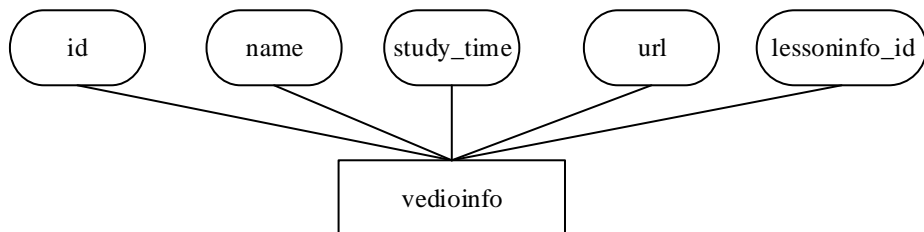


图 4.7 章节视频资源实体图

数据库设计的用户模块的实体图如图 4.8、图 4.9、图 4.10、图 4.11、图 4.12、图 4.13，分别用于记录用户的课程评论信息、用户在网站上的行为信息、用户学习的课程信息、用户的收藏信息，如收藏课程机构和教师、用户的个人信息和用户的邮箱验证码信息；

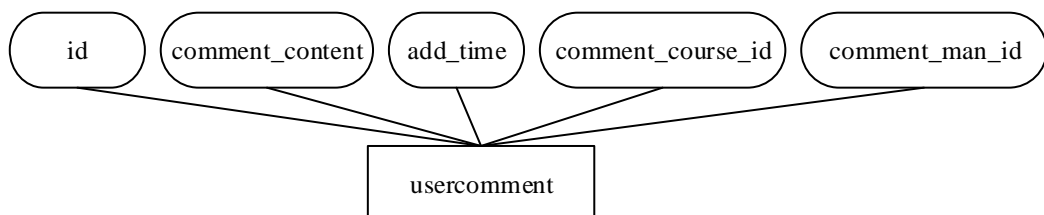


图 4.8 用户评论实体图

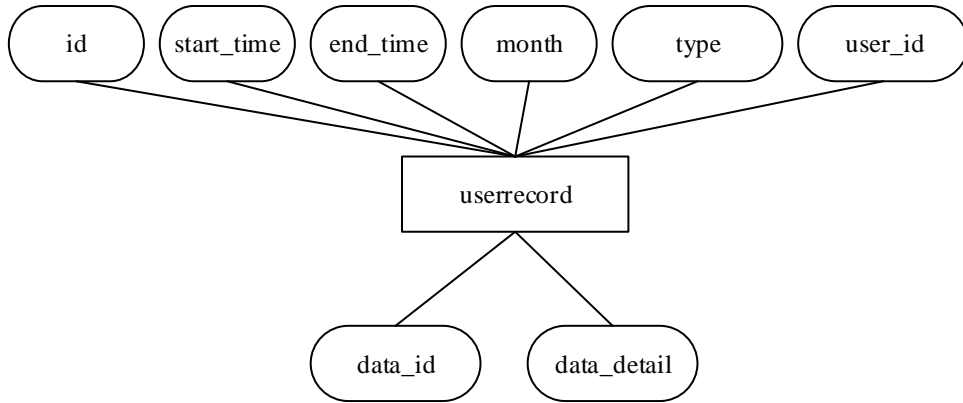


图 4.9 用户行为记录实体图

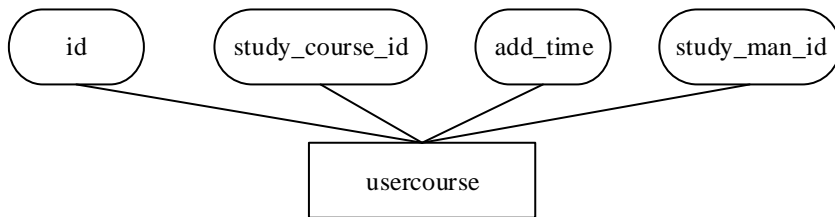


图 4.10 用户课程表实体图

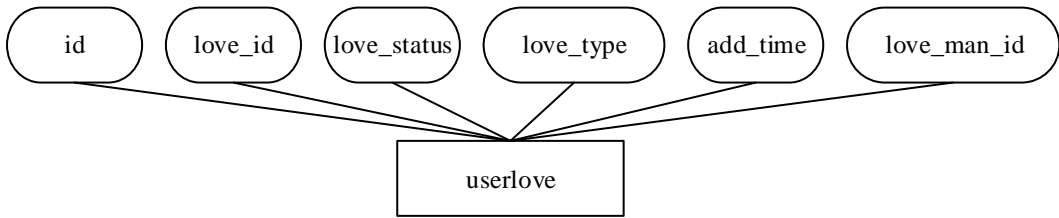


图 4.11 用户收藏实体图

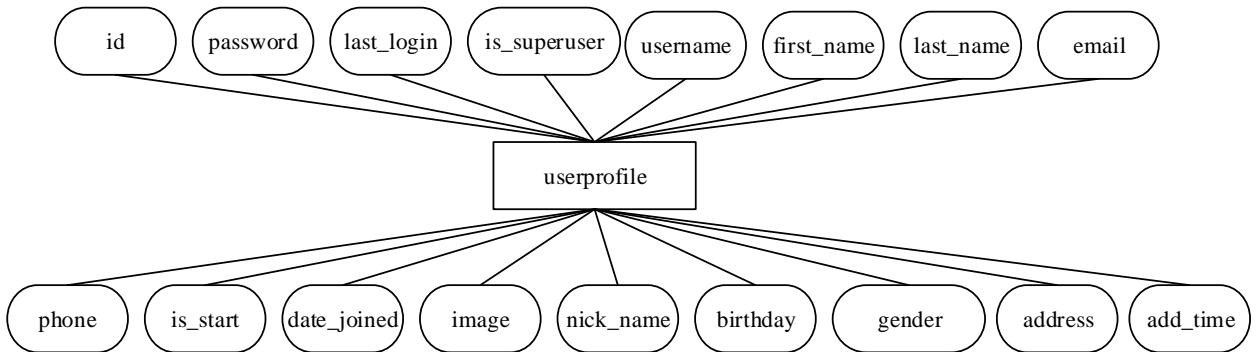


图 4.12 用户实体图

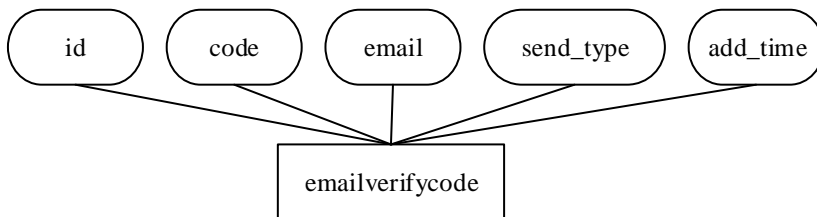


图 4.13 邮箱验证实体图

数据库设计的机构模块的实体图如图 4.14、图 4.15、图 4.16，分别用于记录机构所在的城市信息、机构的详细信息、机构中讲师信息；

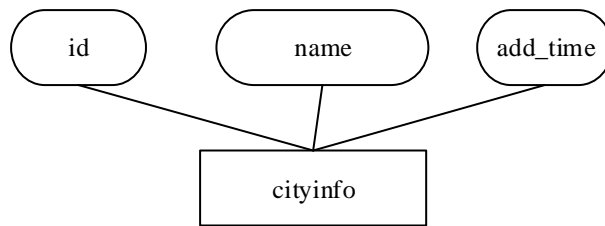


图 4.14 城市实体图

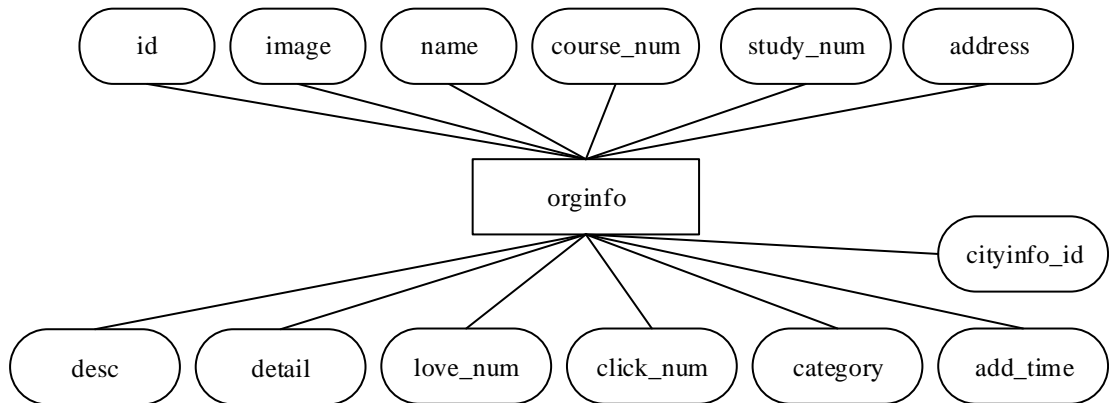


图 4.15 机构实体图

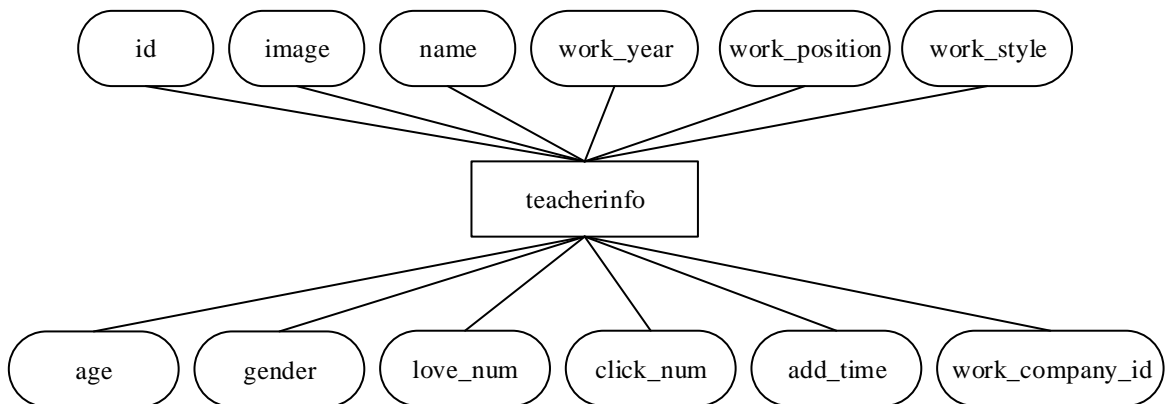


图 4.16 讲师实体图

还有一张数据库表为 bannerinfo，用来维护首页的轮播图信息，如图 4.17 所示：

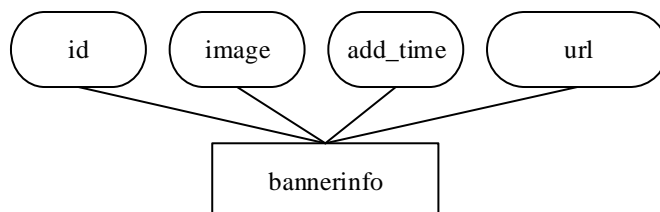


图 4.17 轮播实体图

本文研究的在线学习系统的数据库的 E-R 图如图 4.18 和图 4.19 所示：其中图 4.18 是以用户为中心的数据流、图 4.19 是以课程为中心的数据流。其中章节视频表与章节表是一对多的关系；章节表与课程表是一对多的关系；课程与机构之间、教师与机构之间、课程资源与课程之间都是一对多的关系；用户表与用户评论表、用户表与用户收藏表、用户表与用户行为记录表、用户表与用户课程表之间都是一对多的关系包含相应的物理外键。

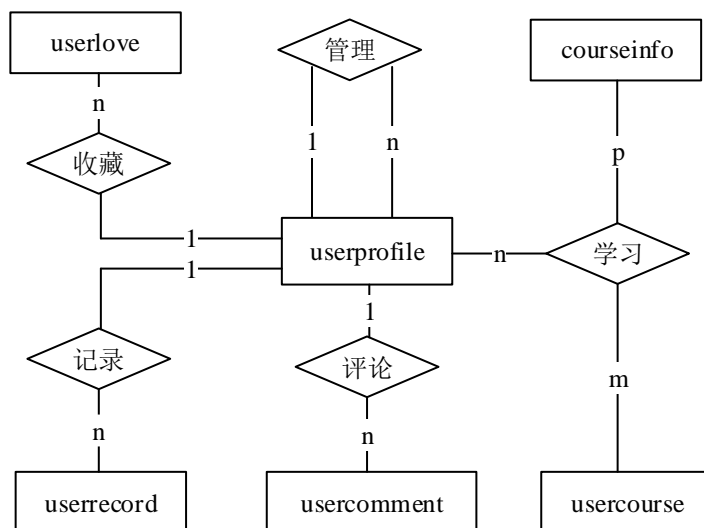


图 4.18 用户为中心的 E-R 图

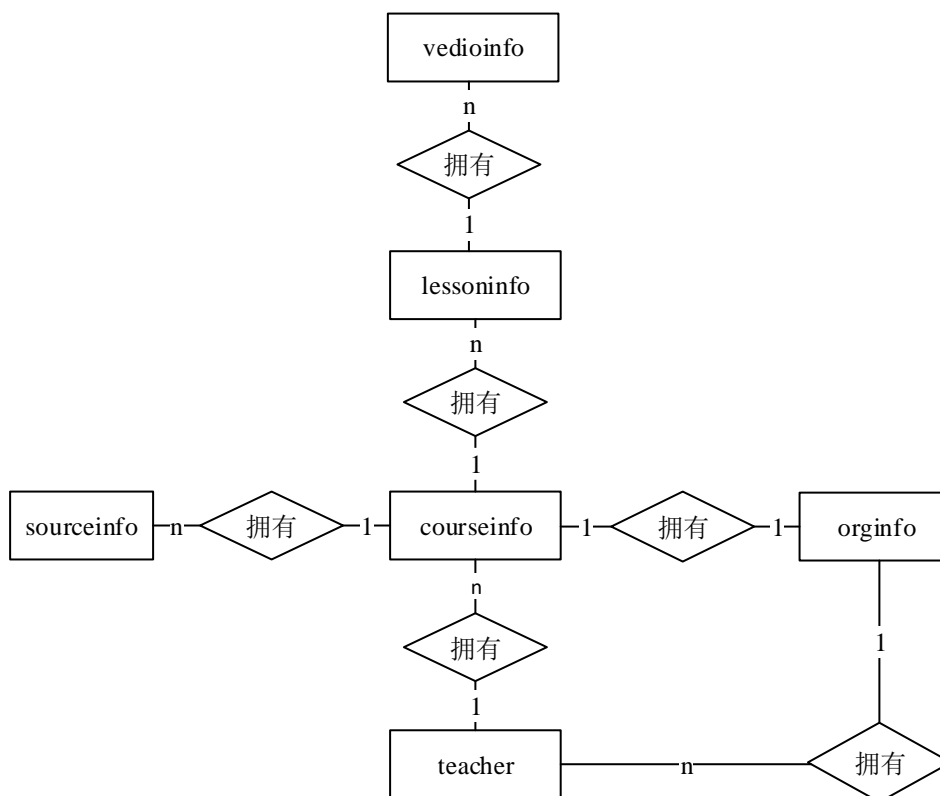


图 4.19 课程为中心的 E-R 图

整个的数据库设计体系中，邮箱验证表与定时任务执行表、session 表是临时数据；而课程资源表、用户收藏表、用户课程表、机构表、用户表、讲师表等都是持久性数据，将

这些数据持久化到数据库中并长久的保留可以为以后数据分析，深度定制化服务、推荐功能提供数据基础。数据库的数据流大体上分为两个部分：以用户表为核心的数据流和以可成为核心的数据流。用户在注册之后会在用户表产生一条数据，围绕这条数据可以产生很多数据，如用户在对课程进行评论时产生用户评论数据、用户收藏课程讲师或者机构时会在用户收藏表中生成数据、用户在页面上的点击事件会在用户行为记录表中产生相应的数据、用户点击课程进行学习会在用户课程表中产生数据；这一条数据流可以通过用户表的主键进行追溯。另一条数据流是以课程核心的数据，通过该课程的主键可以查找到该课程包含的章节信息、课程的课件资源信息、课程的所属机构信息和该课程所属的讲师信息，通过课程章节的主键可以查找到该章节下的所有视频信息，通过教师表的主键可以查找到机构的信息，这样就形成了一条完整数据流，并可以使用课程的主键进行追溯。

4.3 系统架构

本文研究的系统采用了当前业界相对规范的前后端分离的设计模式，使用的 Django 为前台 Vue 构建的页面提供渲染页面的相关数据。在线学习系统分为管理员端的在线学习管理系统和用户端的在线学习系统。

在线学习管理系统包含机构、课程、用户、用户操作、用户行为记录这五个模块的管理管理，数据库是使用非关系型数据库 Redis 和关系型数据库 MySQL 分别对临时数据和持久性数据进行持久化到磁盘操作。各个模块之间独立且没有太大的依赖关系。

用户端的在线学习系统包含课程评论、用户行为操作、用户中心、用户总结、用户注册登录、推荐、用户行为记录、课程学习、搜索、讲师、机构等功能模块，且彼此之间存在很强的依赖关系。用户的行为操作，如用户的对课程、讲师和机构的收藏、对课程的评论等赖于用户注册登录模块，只有成功登录的用户才可以对相关的课程进行评论或者收藏机构、讲师、课程；用户中心和用户行为记录页赖于用户注册登录模块，只有成功登录的用户才能进入用户中心进行相关的操作和对该用户进行行为记录。课程模块是在没有登陆状态下可以浏览，但是不能进入进行深入学习，因此依赖于用户注册登录模块，成功登录的用户才可以进行相关的视频学习，用户总结模块赖于用户注册登录模块和用户行为记录模块，只有登录的用户才能记录到数据信息，并且要对用户的行为信息进行分析才可以给用户发送的用户的总结邮件；用户推荐模块分为热门课程推荐、相关课程推荐和学习过该课程的同学还学习过的课程分析推荐，因此依赖于课程模块、机构模块和用户操作模块；搜索模块分为按课程搜索、按机构搜索和按教师搜索三个方面，因此搜索模块依赖于课程模块、机构模块和讲师模块，分别对搜索到的信息进行展示；最后课程模块依赖于讲师模

块，讲师模块依赖于机构模块，即课程不能独立于讲师存在，讲师不能独立于机构模块存在。

4.4 本章小结

本章详细的对在线学习系统的实现进行分析，分析了相关的流程并给出了流程图；之后介绍了数据库表的详细设计以及实体之间的关系，并给出了相应的 E-R 图，介绍了两条主要的数据流：以课程为核心的数据流和以用户为核心的数据流；再其次介绍了，系统采用了前后端分离的架构，并详细的描述了各个功能模块以及各个功能模块之间的依赖关系。在介绍以上的技术和功能的基础上，诠释了本文研究系统的理论支持、具体功能模块的分解和整体的设计实现思路。

第五章 在线学习系统详细设计与实现

5.1 后台管理系统的实现

本文研究的在线学习系统的管理系统的后台是使用以快速开发并具有显著的安全性的著称的 Django 框架进行开发设计的。结合 Django 框架的自身特点和本文研究系统所需最终决定使用第三方插件 Xadmin 对在线学习系统的管理系统进行快速开发：不需要大量的冗余代码只是简单的配置就生成了一个具有很强大功能且界面友好的管理系统。配置生成的系统具有登录页面和后台管理页面，后台管理又包含机构模块的管理，可以对机构的城市信息、机构信息、讲师信息进行增加、修改、删除和通过不同的条件进行过滤查询等管理操作；还包含用户操作管理模块，可以对不同用户的收藏信息，即课程收藏、讲师收藏、机构收藏，进行包含但不限于增加、按用户与时间和类别等进行不同的过滤查找或导出为 Excel 等诸多格式的文件，对用户的学习信息进行基础的增加、修改、删除操作和按照用户、课程或者时间对用户的学习情况进行过滤、导出数据、按条件搜索等管理操作，可以对用户课程的评论可以进行多条件过滤查找等，对一些不真实的虚假评论和无关内容的或涉及到非法色情暴力的评论进行删除操作；其次在线学习系统的管理系统还包含用户管理模块，可以对用户信息进行包含但不限于增删改的操作的基本操作和按条件过滤、导出为 Excel、JSON 等数据格式的文件，还可以对首页展示的轮播图进行相应的管理。同时该管理系统还包含用户行为记录的管理，可以清楚的了解到用户在界面上的每一个界面的停留时间和点击事件，并对这些数据进行包含但不限于增删改的基本管理操作，可以按照类型、用户、月份这些条件进行过滤、导出为 Excel、JSON、XML 等格式的文件；最后还包含课程管理模块，对课程信息进行包含但不限于增删改的基本管理操作，可以按照课程名、学习人数、收藏数、类别等条件进行过滤查找和导出为常见的文件，对课程的章节信息进行包含但不限于增加、修改、删除的管理操作，可以导出为常见的文件和按照章节名称、所属课程等条件进行过滤查找，对章节的视频信息进行包含但不限于增删改的基本管理操作和导出为常见文件，并且可以视频时长、所属章节进行过滤操作，对该课程的课件资源进行包含但不限于增删改的基本管理操作，可以按照名称、所属课程进行相应的过滤查找操作和导出为常见的文件；在线学习系统管理系统还可以对后台的管理员操作进行详细的日志记录。具体页面如图 5.1 所示：



图 5.1 管理员端管理系统 UI 图

下面提供具体的使用方法和配置的代码，详细的讲述了如何生成该管理系统，将 Xadmin 插件下载之后放到项目下作为一个独立的模块，并且在本项目的配置文件安装这个模块配置路由映射之后进行 Python 的数据迁移就可以成功的完成模块的引进；在之后就是对不同的模块做后台管理的配置，在每个模块下都新建 Xadmin.py 程序，在其中配置显示内容、可以进行搜索的字段和可以进行筛选的字段并注册模块，最后通过命令创建一个管理员就可以实现对配置内容的管理，包含但不限于简单的增加、修改、删除、按条件过滤和导出为本地文件的功能。如下是管理员端的核心实现代码，其中以每个模块的 Xadmin.py 仅以用户操作模块的代码为例。

```

01. # 1. setting.py 文件中的 INSTALLED_APPS 加入 xadmin:
02. INSTALLED_APPS = [
03.     'DjangoUeditor',
04.     'xadmin',
05. ]
06. # 2. 在本项目的 url.py 文件中添加 url 映射:
07. urlpatterns = [
08.     url(r'^xadmin/', xadmin.site.urls),
09. ]
10. # 3. 在控制台同步下数据库，并创建用户
11. python manage.py makemigrations
12. python manage.py migrate
13. python manage.py createsuperuser
14. --username=zack --email=zzhang_xz@163.com

```

```

01. # 4. 创建不同模块的 xadmin.py 文件：这里只是以用户操作做为示例，其他模块一样。
02. import xadmin
03. from .models import *
04. class CityInfoXadmin(object):
05.     list_display = ['name', 'add_time']
06.     model_icon = 'fa fa-bath'
07. class OrgInfoXadmin(object):
08.     list_display = ['image', 'name', 'course_num', 'study_num',
09.                     'love_num', 'click_num', 'category', 'cityinfo']
10.     model_icon = 'fa fa-gift'
11.     style_fields = {'detail': 'ueditor'}
12. class TeacherInfoXadmin(object):
13.     list_display = ['image', 'name', 'work_year',
14.                     'work_position', 'click_num', 'age', 'gender', 'love_num']
15. xadmin.site.register(CityInfo, CityInfoXadmin)
16. xadmin.site.register(OrgInfo, OrgInfoXadmin)
17. xadmin.site.register(TeacherInfo, TeacherInfoXadmin)

```

除此之外，在线学习系统的管理系统还包含第三方的 DjangoUeditor 的使用，具体使用的方法一样，就不在此展开讨论了。

5.2 用户模块功能

本文研究的在线学习的内容，是一种长效的服务机制而不是一次性使用的网站。因此，需要用户可选择的提供一些个人信息并记录用户的收藏、学习信息，在此基础上才能更加有效精准的提供长期的服务，例如每个月月初向用户发送上个月的学习情况总结并给出学习的建议等。还有就是在如今的时代可以通过丰富的数据来对用户进行分析并做出相关的推荐，因此就需要在线学习系统的用户访问的数据；然而这一切都需要验证用户的邮件确实属于本人，才能不给其他人带来烦恼的垃圾邮件，所以就在用户注册的时候验证邮箱的信息。

因此在设计用户注册的时候，添加了到邮箱激活账户的功能，如果用户不进行账户的激活，则不能成功的登录本网站。如下是用户模块的核心实现代码：

```

01. # 1. 配置 URL
02. urlpatterns = [
03.     url(r'^user_register/$', user_register, name='user_register'),
04.     url(r'^user_active/(\w+)/$', user_active, name='user_active'),
05. ]
06. # 2. 注册账号
07. def user_register(request):
08.     user_register_form = UserRegisterForm(request.POST)
09.     if user_register_form.is_valid():
10.         email = user_register_form.cleaned_data['email']
11.         password = user_register_form.cleaned_data['password']
12.         user_list = UserProfile.objects.filter(Q(username=email) | Q(email=email))
13.         if user_list:
14.             return JsonResponse({
15.                 'errMsg': '用户已经存在'
16.             })
17.         else:
18.             ...
19.             return JsonResponse({
20.                 'errMsg': '请去邮箱激活账号'
21.             })
22.         else:
23.             return JsonResponse({
24.                 'errMsg': '输入信息格式错误'
25.             })
26.
27. # 3. 激活账号
28. def user_active(request, code):
29.     if code:
30.         email_ver_list = EmailVerifyCode.objects.filter(code=code)
31.         if email_ver_list:
32.             email_ver = email_ver_list[0]
33.             email = email_ver.email
34.             user_list = UserProfile.objects.filter(username=email)
35.             if user_list:
36.                 ...
37.                 return JsonResponse({
38.                     'errMsg': 'ok'
39.                 })

```

已经成功登录的用户可以进入个人中心，可以对个人的收藏如机构、讲师、课程等信息进行查看和删除操作；可以对现在或之前已经学习过的课程进行查看；还可以改变个人的相关资料等。详细的页面如图 5.2 所示：



图 5.2 个人中心的 UI 图

个人中心的核心实现代码如下，这里只展示了前端的 Vue 课程相关的代码：

```

01. <div v-for="(course, index) in course_list" >
02.   <div class="des">
03.     <a @click="$router.push(`/courses/course_detail/${course.pk}`)" >
04.       <h2>{{course.fields.name}}</h2> </a>
05.       <span class="fl"> 课时: <i class="key">{{course.fields.study_time}}</i></span>
06.       <span class="fr">学习人数: {{ course.fields.study_num }}</span>
07.     </div>
08.     <div class="bottom">
09.       <span class="browse fl" title="浏览次数">{{course.fields.click_num}}</span>
10.       <span class="star fr" title="收藏人数">{{course.fields.love_num}}</span>
11.     </div>
12.   </div>
13.   export default {
14.     data() {
15.       course_list:[],
16.       showComponent: false,
17.     },
18.     mounted() {
19.       axios({ url: '/api/users/user_course/', method: 'GET', })
20.         .then((response)=>{
21.           var res = response.data
22.           this.course_list = JSON.parse(res.course_list)
23.           this.showComponent = true
24.         }).catch((err)=> {
25.           console.log(err)
26.         })
27.     },
28.   };

```

5.3 课程模块功能

用户端的在线学习系统的课程模块，按照最新课程、学习人数或浏览人数等条件进行过滤查找；还包含了热门课程推荐、以及与本课程相关的课程推荐还有学习过该课程的用户还学习过的课程推荐；从课程列表页面进入课程的详情页，课程详情页面可以查看到该课程所述的机构，并可可以决定是否收藏该课程或者收藏该机构，收藏之后可以在用户的个人中心对收藏的数据进行查看；之后可以点击开始学习，在此处若是没有成功登录的用户会被网站重新导向到网站的登录页面进行登录之后网站会自动的跳转到之前课程的学习页面进行学习；课程学习页面，包含该章节、课程、课件、评论等功能元素。具体的课程学习页面如图 5.3 所示：



图 5.3 课程评论 UI 图

下面详细的介绍课程的评论功能和推荐功能：课程的推荐功能原理是通过查询学习过该课程的用户，并获取这些用户的所学习的课程之后进行一定的数据处理，最终经这些课程向用户进行推荐；课程的评论功能是查取该课程的最新的数十条评论进行展示，且已经成功登录的用户可以在法律允许的内优雅的发表评论。如下是课程评论的核心实现代码：

```

01. @login_decorator
02. @cache_page(60 * 24 * 60)
03. def course_video(request, course_id):
04.     if course_id:
05.         course = CourseInfo.objects.filter(id=int(course_id))
06.         usercourse_list = UserCourse.objects
07.             .filter(study_man=request.user, study_course=course)
08.         if not usercourse_list:
09.             ...
10.             # 维护机构的学习人数
11.             usercourse_list = UserCourse.objects
12.                 .filter(study_man=request.user).exclude(study_course=course)
13.             course_list = [usercourse.study_course for usercourse in usercourse_list]
14.             org_list = list(set([course1.orginfo for course1 in course_list]))
15.             # 学过该课的同学还学过什么课程
16.             usercourse_list = UserCourse.objects.filter(study_course=course)
17.             user_list = [usercourse.study_man for usercourse in usercourse_list]
18.             usercourse_list = UserCourse.objects
19.                 .filter(study_man__in=user_list).exclude(study_course=course)
20.             courselist = list(set([usercourse.study_course for usercourse in usercourse_list]))[:6]
21.             course_sources = SourceInfo.objects.filter(courseinfo=course)
22.             lesson_list = LessonInfo.objects.filter(courseinfo=course)
23.             for lesson in lesson_list:
24.                 video_list = VideoInfo.objects.filter(lessoninfo=lesson)
25.                 for video in video_list:
26.                     a = {'name': video.name, 'url': str(video.url), 'study_time': video.study_time}
27.                     videos.append(a)
28.                 lessons.append({'name': lesson.name, 'video': videos})
29.             return JsonResponse({
30.                 'course': serializers.serialize("json", courseQuerySet),
31.                 'course_list': serializers.serialize("json", courselist),
32.                 'course_sources': serializers.serialize("json", course_sources),
33.                 'lesson_list': json.dumps(lessons),
34.             })
35. # 用户评论
36. def course_comment(request, course_id):
37.     if course_id:
38.         course = CourseInfo.objects.filter(id=int(course_id))
39.         all_comments = course.usercomment_set.all().order_by('-add_time')[:10]
40.         for comment in all_comments:
41.             user = UserProfile.objects.filter(id=comment.comment_man_id)[0]
42.             comment_info = {'add_time': str(comment.add_time), 'comment_content':
43.                 comment.comment_content, 'image': str(user.image), 'nick_name': user.nick_name}
44.             comment_vo.append(comment_info)
45.         return JsonResponse({
46.             'course': serializers.serialize('json', courseQuerySet),
47.             'comment_vo': json.dumps(comment_vo),
48.         })

```

如代码中所示，需要在登录状态才能进对视频进行学习和收藏等操作，并在完成系统重定向到的登录之后再次回跳到之前的页面，因此需要自己定制自己的装饰器。而代码中的@cache_Page 注解是使用 Redis 对数据进行了缓存未来提高用户的使用体验。

5.4 机构模块功能

用户端的在线学习系统具有机构模块，用户可以使用机构的类别、机构的所在地、机构的学习人数、机构的课程数等条件进行过滤查找；还绘制了满足当前条件的机构并按照收藏数的排行榜方便用户获取优质的机构信息；点击进入机构的首页，包含机构的课程、机构的讲师、机构的简介等信息方便用户对该机构有全面的认知。完成的机构页面如图 5.4 所示：



图 5.4 机构展示 UI 图

这里面涉及到的搜索、排序、按不同的条件进行过滤，并将获取到的机构列表进行分页等操作，详细的实现过程与课程模块的功能基本类似。

5.5 用户行为记录模块功能

用户端的网络学习系统在当今的社会环境中就应该适应潮流的发展记录海量的用户行为数据为以后的数据分析、优化网站性能、提供优质的定制化服务等铺垫坚实的数据基础，如每个月的月初向用户发送该用户在本网站的学习情况总结等。因此需要大量的对用户在网站上的点击和页面的停留时间等行为进行翔实的记录。

因此在研究 Django 框架之后最终采用中间件的方式在后台对用户的点击行为进行记录，分为点击了课程、点击了机构、登录时长三个方面和记录用户在每个界面的停留时间；

只有成功的通过 Django 自身框架的中间件和用户自定义的中间件处理的请求才可以被分配到不同的视图层进行相应的逻辑处理，否则就会报错，所以本文自己手动的写了一个中间件对所有的前端页面的请求进行处理。

但是这里面存在一个技术性的难点，简单的描述为当用户不进行任何页面操作或直接关闭浏览器时在后台中间件代码难以触发，就不能够记录到用户的离开时间，即使 session 失效后台的中间价也难以得知。在查阅很多的信息之后最终采取了使用 Redis 记录所有的在线用户和在后台代码中书写了一个定时任务去判断这些用户有没有离开，即 session 失效，当用户的 session 失效时则认为这个用户退出本网站并将当前时间数据保存到记录表中；这里面就要考虑何时的 session 时间来确保用户的网站使用体验。当然这里的 Redis 使用有效的减轻了数据的压力。具体的定时任务的核心代码和 Django 中间件的核心代码如下：

```

01. # 中间件
02. class OnlineStatusMiddleware(MiddlewareMixin):
03.     def process_request(self, request):
04.         cache_key = 'userinfo_'+request.user.username
05.         cache_request_path_key = "request_path" + request.user.username
06.         user_expire = cache.ttl(cache_key)
07.         if request.user.is_authenticated():
08.             now = datetime.now()
09.             lastPath = cache.get(cache_request_path_key)
10.             this_path = request.path # 将上一次的 path 缓存到 redis
11.             if 'get_user' in this_path:
12.                 cache.expire(cache_request_path_key, settings.REDIS_TIMEOUT)
13.             else:
14.                 cache.set(cache_request_path_key, this_path, settings.REDIS_TIMEOUT)
15.             this_user = request.user
16.             userinfo = cache.get(cache_key) # 缓存整个 user 的信息
17.             if not userinfo:
18.                 cache.set(cache_key, request.user, settings.REDIS_TIMEOUT)
19.             else:
20.                 cache.expire(cache_key, settings.REDIS_TIMEOUT)
21.             # 记录 login/logout
22.             userrecord = UserRecord.objects
23.                 .filter(user=request.user, type=1).order_by('-start_time')
24.             if not userrecord:
25.                 ur = UserRecord()
26.                 ...
27.                 ur.save()
28.             elif 'get_user' in request.path: # F5
29.                 ...
30.             # 记录用户所有的点击行为
31.             if not lastPath or lastPath != this_path:
32.                 ...
33.         return None

```

```

01. # 定时任务
02. def job_session_invalid():
03.     all_cache_key_in_cache = cache.iter_keys("userinfo_*")
04.     for cache_key in all_cache_key_in_cache:
05.         expire_time = cache.ttl(cache_key)
06.         if expire_time < 60 * 4 + 58:
07.             userinfo = cache.get(cache_key)
08.             userrecord = UserRecord.objects
09.                 .filter(user=userinfo, end_time=None).order_by('-start_time')
10.             cache.delete_pattern(cache_key)
11.             cache_request_path_key = "request_path" + userinfo.username
12.             cache.delete_pattern(cache_request_path_key)
13.             for record in userrecord:
14.                 record.end_time = datetime.now()
15.                 record.save()
16.         else:
17.             pass
18.     else:
19.         pass

```

在这个模块中还有一个定时任务就是每个月的月初向用户发送一封定时邮件总结上个月在本网站的学习情况，包含登录的数量、在线的总时间、点击的课程数量、点击的课程机构数量等信息，如下是月初的定时邮件的核心代码：

```

01. def job_send_summary_email():
02.     # 每个月月初发送给用户上个月的学习总结
03.     all_users = UserProfile.objects.all()
04.     email_month = datetime.now().month - 1
05.     try:
06.         for user in all_users:
07.             userrecorded = UserRecord.objects
08.                 .filter(month=email_month, user=user)
09.             if not userrecorded:
10.                 send_email_code(user.email, 3)
11.             else:
12.                 learning_times = userrecorded
13.                     .filter(type=1).count()
14.                 learning_courses_time = userrecorded
15.                     .filter(type=2)
16.                 learning_time = 0
17.                 for record in userrecorded: learning_time = learning_time+
18.                     (record.end_time-record.start_time).seconds
19.                 send_email_code_job(user.email, learning_times,
20.                     learning_courses_time, learning_time)
21.     except Exception as e:
22.         print('发送用户总结任务时出错' + e)

```

5.6 系统测试

本文研究的在线学习系统包含用户端的在线学习系统和管理员端的在线学习管理系统，作为一个独立的网站项目应该在详细的测试后才能进行之后的部署与使用。而网站测试目标是保证功能完整、无明显的漏洞、且流程逻辑完整、保证良好的用户体验，且覆盖

尽量多的功能点。因此本文会对系统的功能点进行详细的测试，这里就几个主要的功能点进行详细的描述。

后台管理系统的测试的目的是可以正确的管理所有的用户数据，测试的过程为注册一个用户后在界面上进行学习，之后进行后台管理系统查看数据；预期是在数据库内可以查阅到该用户的浏览数据，结果是确实查到了该用户产生的数据流。在后台删除或修改数据课程信息，预期用户端将不在展示这个数据，结果到了用户端的系统登录后就不在可以查看到这个信息。还对管理员的每一个功能点都进行了相应的测试，而都符合之前的预期。

用户模块测试了在注册时发送激活账户的邮件，点击预期的邮件链接并在数据表中产生一条 `is_start` 字段为 `False` 的记录，且不能成功的网站登录，点击链接之后用户表中的是否激活字段为 `True` 且可以成功的登录网站，测试的结果完全符合预期。具体如图 5.5、5.6、5.7 所示：

add_time	is_start
2018-09-21 15	0

图 5.5 未激活时的数据库图

图 5.6 未激活时的登录 UI 图



图 5.7 注册激活邮件图

对用户行为记录模块进行相关的测试，预期是当用户成功的登录本网站是会在数据库的用户行为记录表中产生一条只有 start_time 没有 end_time 且 type 为 1 的数据，用户在页面上进行学习等点击行为都会在用户行为记录表 userrecord 中产生数据，如图 5.8 所示。

210	2019-05-21 15:57:41.404/04	2019-05-21 15:59:20.744216	5	2	29	1 /
212	2019-05-21 21:25:56.937591	(Null)	5	1	29	(Null) (Nu
213	2019-05-21 21:26:57.126491	(Null)	5	2	29	1 /

图 5.8 用户操作行为记录的数据库图

在等待一定的时间 session 过期之后，这两条数据记录都会填写完整，符合预期，如图 5.9 所示。

212	2019-05-21 21:25:56.937591	2019-05-21 21:47:08.573157	5	1	29	(Null) (Nu
213	2019-05-21 21:26:57.126491	2019-05-21 21:47:08.364715	5	2	29	1 /co

图 5.9 定时任务执行的数据库图

5.7 本章小结

本章对本文研究的在线学习系统及其后台管理系统的具体实现方法进行了翔实的叙述，着重介绍了管理系统的配置实现和用户模块以及课程机构模块的功能介绍和实现思路，并给出了其中核心部分的代码；还诠释了用户的行为记录的思考和实现的过程也给出了其中的核心实现代码，以及交代了本文中的两个定时任务；最终对核心功能代码的测试工作，且符合测试的预期。

第六章 结论与展望

6.1 本文主要成果

经历这次的毕业设计之后个人感觉无论在学术上还是个人成长方面都取得一些积极的成果。在经过认真且深入的学习研究了 Django 和 Vue 等技术的基础上进行了大量的中英文文献的阅读，对在线学习系统及相关的内容进行了细致的了解。随着互联网技术的不断革新，对本文研究的系统的现今发展水平和涉及到的技术都有了直观的认知，如现在比较常见而被使用的 SSH 框架还是存在着学习成本较大难以快速开发等诸多缺陷，而不能适应如今的海量数据的高并发的发展趋势。因此在本文研究的系统中使用了前后端分离的架构达到前端视图化、后台数据化和组件化编码等现今比较有优势的理念与思想，如此可以实现快速开发与部署和降低运维成本。比如其中的组件化编码就可以大大的提高代码的复用率从而可以为快速开发提供理论支持，后端的数据化就会在不断的积累的过程中最终沉淀汇聚成中台，如此就为后端的快速开发提供坚实的基础和理论支持，从而使得软件系统具有不断迭代和升级的能力真正的保证了软件的质量，最终从诸多理论中脱颖而出。

在本次的研究在线学习系统搭建的过程中查询了现在的常用的技术栈分析这些技术的优缺点后，构建了多种选择方案并进行对比筛选并最终决定了使用前后端分离的架构来实现前端页面化和后端的数据化的方案，后端采用以快速开发著称的 Django 框架结合前端的新兴的 Vue 框架并结合与之配套的插件进行开发。其次，还充分的考虑到了本网站的社会影响，可以对社会的教育行业奉献绵薄之力使得学生可以更加愉快的学习知识；当然作为知识密集型的系统，其中的视频课程的版权都被充分的考虑到使得其在现有的法律的框架内，使得他人的权益不受侵害。最后，对系统的可运行性、数据准确性、流程完整性等方面进行了测试，得出了本文系统在解决在线教育的搭建上具有很大的优势，适合快速开发部署。

6.2 未来工作展望

在本次在线学习系统搭建的研究过程中对 Django 和 Vue 等相关技术进行学习使用有了清晰的认识。因此认识到本文系统的后台管理系统和前端从后台获取数据有一定时延等方面还是存在一定的缺陷，而且只是利用 Django 框架本身去进行并发，在这个方面没有自己进行优化并提高流量吞吐。这个会在现实的服务器上具有很大的性能要求。

因此，在未来的一段时间内会使用以高并发 IO 密集且处理少量业务逻辑著称的 NodeJS 在前端内继续搭建一个前端的 Node 服务器，对从后台获取到的 JSON 数据进行相关的处理直接生成 HTML 代码后交给浏览器显示。这样就会减轻浏览器并不擅长处理数据的劣势，从而达到快速响应的用户提高用户的使用体验。其次就是研究并发的优化策略做到最大化的利用服务器性能和优化数据库的底层数据的查询，做到适应时代的并发。并且在用户推荐方面使用协调过滤算法，协同过滤推荐系统(CFRS)作为一种比较成功的推荐系统^[20]，从而为用户推荐更加优质的内容。

参考文献

- [1] 封宇, 陈宁江. 基于 MVVM 架构的移动 Web 前端展示方案[J]. 计算机与现代化, 2014(11):1-4.
- [2] Xiaoli Liu,Xue Li,Xiaolin Chen.Location correction technique based on mobile communication base station for earthquake population heat map[J].Geodesy and Geodynamics,2018,9(05):388-397.
- [3] 教育部基础教育司 2018 年工作要点[J].基础教育参考,2018(06):3-5.
- [4] 孔德民. 基于 Python 开发预警机系统检测设计与研究[D].哈尔滨理工大学,2017:1-2.
- [5] 朱二华.基于 Vue.js 的 Web 前端应用研究[J].科技与创新,2017(20):119-121.
- [6] 麦冬, 陈涛, 梁宗湾. 轻量级响应式框架 Vue.js 应用分析[J]. 信息与电脑(理论版), 2017 (7):58-59.
- [7] 薛耀伟. 基于 Django 框架管理界面自动生成模块的设计与实现[D].哈尔滨工业大学,2014:1-2.
- [8] 赵海丹. 基于 LNMP 的智能农业服务平台的研究[D]. 2015:1-2.
- [9] 马文龙,朱好晴,蒋德钧.Key-Value 型 NoSQL 本地存储系统研究[J].计算机学报,2018,41(08):1722-1751.
- [10] 马豫星. Redis 数据库特性分析[J]. 物联网技术, 2015(3):105-106.
- [11] 徐頔, 朱广华, 贾瑶. 基于 VueJs 的 WEB 前端开发研究[J]. 科技风, 2017(14):69-69.
- [12] B Gao , S Zhang , N Yao . A Multidimensional Pivot Table Model Based on MVVM Pattern for Rich Internet Application[J].International Symposium on Computer , 2012:1-2
- [13] 秦书荣, 李新. 响应式网页设计的原型研究[J]. 电子技术与软件工程, 2015(22):9-10.
- [14] 邓雯婷. 基于 Vue.js 构建单页面 GIS 应用的方法研究[J]. 科技创新与应用 , 2018, No.234(14):11-13+16.
- [15] 张云苑. JavaScript 在动态网页设计中的应用[J]. 科技信息, 2007(5):23-24.
- [16] 孙一笑,张玉军,孙宇成.基于 WebAPI 前后端完全分离的软件开发模式[J].信息与电脑(理论版),2019(06):96-97.
- [17] 范开勇,陈宇收.MySQL 数据库性能优化研究[J].中国新通信,2019,21(01):57.
- [18] Yadong Guo. Study on High-performance Distributed Cache Architecture based on Redis[A]. Information Engineering Research Institute,USA.Proceedings of 2012 2nd International Conference on Advanced Materials and Information Technology Processing(AMITP 2012) Volume 34[C].Information Engineering Research Institute,USA:Information Engineering Research Institute,2012:5.
- [19] Ming Xu,Xiaowei Xu,Jian Xu,Yizhi Ren,Haiping Zhang,Ning Zheng. A Forensic Analysis Method for Redis Database based on RDB and AOF File[J]. Journal of Computers,2014,9(11):1-5.
- [20] Guangqi Wen,Chunmei Li. Research on Hybrid Recommendation Model Based on PersonRank Algorithm and TensorFlow Platform[J]. Journal of Physics: Conference Series,2019,1187(4):1-4.

致 谢

本文研究系统是在导师王进老师近三四个月的耐心的指导下完成的，在这次毕业设计中让自己更加的充实忙碌的学习到各种知识在知识的海洋中徜徉。对自己的有了深刻的全面的认识，在严谨教学为人可亲的教师和互帮互助嬉戏打闹的室友的帮助下，有良多感悟。为以后的人生职业规划方面打下了坚实的基础，并逐渐的掌握了一些自我成长的方法和学习新知识的途径与一些学习方法，可以渐渐地适用社会为今后的步入社会提供了一个有效的缓冲。其次就是学会了软件行业相关技术问题的描述和掌握如何与他人有效的沟通进而了解社会相关行业的新动态，能够清晰的表述自己的观点和听懂别人的叙述并做出适合的回应，建立平和稳定的长效联系。

这里要感谢王进老师的耐心指导与悉心的教诲和我亲爱的室友练顺的无私帮助，就像为迷途中的我照料了希望的灯塔，从而有了在大学阶段的人生小目标并为之而努力拼搏。在这些帮助和老师的建议下，我对本文研究的系统进行了一次又一次的改善并最终形成，因此在次表达对王进老师和练顺室友的最崇高的敬意和由衷的感谢。王老师的敬业精神和专业知识素养与无私的奉献精神深深地打动了我的内心，在我内心中留下来真正学者的风范，令人油然而敬。

另外我还要感谢我所在的南通大学，感谢过去四年中所有传授我宝贵知识有教无类的老师们，感谢我的家人以及朋友们，感谢所有关心和帮助过我的人。除此之外，我还想感谢国家，为我提供了安全舒适的生长环境，可以无忧无虑的野蛮生长，因此为自己生长在这个伟大的时代伟大的过度而庆幸。