

西安交通大学

硕士学位论文

基于 RTMP 流媒体技术的围棋直播教学系统的设计与实现

学位申请人：李尧

指导教师：赵英良教授

类别（领域）：工程硕士（软件工程）

2018 年 4 月



# **Design and Implementation of Go Live Teaching System Based on RTMP Streaming Media Technology**

A thesis submitted to  
Xi'an Jiaotong University  
In partial fulfillment of the requirement  
for the degree of  
Master of Engineering

By  
Yao Li  
Supervisor: Yingliang Zhao  
(Software Engineering)  
April 2017



论文题目：基于 RTMP 流媒体技术的围棋直播教学系统的设计与实现

类别（领域）：工程硕士（软件工程）

学位申请人：李尧

指导教师：赵英良教授

## 摘 要

伴随着基于互联网和人工智能的人机对弈的发展，围棋作为一项有着悠久历史的益智类游戏，吸引了越来越多的人投入时间和金钱来学习。但是，围棋的教学模式过于单一，导致围棋运动的推广方式较为单一，围棋爱好者。通过互联网的高效传播性来解决围棋教学的地域性限制对于围棋的学习与推广十分有意义。

针对围棋教学的思考，本文对基于互联网模式的围棋直播教学系统做出了分析。首先，通过介绍 Nginx、uWSGI、Django 来说明教学系统的后端服务器架构，将业务逻辑与流媒体两类请求分发到不同功能的服务器。其中，基于 RTMP 的流媒体服务器为围棋的直播课堂提供了底层支持，课堂上的教学互动则依赖于 Websocket 的长连接技术。本文通过需求描述和对系统的交互模型、结构模型、行为模型的分析，清晰地展现了围棋直播教学系统中应包含的学生与教师用户管理、在线课堂学习、用户升级、创建学校、开设课堂、校园管理与课程管理等需求。在需求分析的基础之上，本文还说明了围棋直播教学系统的 B/S 架构以及系统整体的模块结构划分。通过对基本信息与虚拟钱包、创办学校与开设课堂、围棋资源与在线课堂以及校园管理与课程管理四个具体模块内部的静态结构和类间交互的描述，展示了 Django 框架下系统内各主要类对象的内部结构以及系统内部的信息流通过程。之后，描述了系统开发所需的软硬件环境、服务器的配置，包括 Nginx、uWSGI、流媒体服务器的配置，展示了各功能模块内的主要类与函数的代码，测试包括测试环境、功能性测试以及系统的非功能性测试。

在实验中，本文所提出的基于 RTMP 流媒体技术的围棋直播教学系统可以很好的帮助教师用户建立围棋网上学校，帮助学生用户参与直播课堂进行围棋学习，较好的达到了预期效果。

关 键 词：RTMP;流媒体视频直播; Nginx;uWSGI; Django;Websocket

论文类型：应用研究



**Title: Design and Implementation of Go Live Teaching System Based on RTMP  
Streaming Media Technology**  
**Professional Fields: Software Engineering**  
**Applicant: Yao Li**  
**Supervisor: Prof. Yingliang Zhao**

## ABSTRACT

With the development of man-machine game which is based on the Internet and artificial intelligence, go sport ,as a puzzle game with the long history,has attracted more and more people spend time and money to go to learn.However, the teaching mode of Go is too monotonous, and the promotion of Go is extremely limited.So,it is of great significance for the promotion of go learning to solve the regional limitation of go teaching through the efficient dissemination of the Internet.

This thesis makes an analysis of Go teaching system based on Internet mode.First,this thesis illustrates the back-end server architecture of the teaching system by introducing Nginx, uWSGI and Django, the business logic and streaming media are distributed to the servers of different functions.Among them, the streaming media server based on RTMP provides the low-level support for live classes of go, and the teaching interaction in the classroom relies on the long connection technology of Websocket.According to the requirement description and the analysis of the interactive model,static model and behavior model,clearly shows the user management of students and teachers, online classroom learning,user upgrading,school creating,classroom opening and management that online game teaching needs.On the basis of demand analysis, this thesis also explains the B/S architecture of go teaching system and the division of the module structure of the system as a whole. Through the detailed analysis of the static structure of the module internal,such as basic information and virtual account,establishing schools and opening classes, go resources and participating in class, management of school and classroom,as well as the description of the interaction between classes,this thesis shows the internal structure of the main class object system and the information flow process within the system in the Django framework .After that, it describes the software and hardware environment needed for the system development, the server's configuration, including Nginx, uWSGI, streaming media server configuration, display the function of each module in the main class and function of the code, test environment, including functional testing, and system of non-functional testing.

In the experiment and practical use, the go live teaching system based on RTMP streaming media technology which proposed in this thesis can help build go online school for teacher users and help student users to participate in the class to study,overall,this system that get a good achievement as the expected effect.

**KEY WORDS:** RTMP;live streaming video; Nginx;uWSGI; Django;Websocket

**TYPE OF THESIS:** Application Research



# 目 录

1 绪论.....	1
1.1 选题意义及应用背景.....	1
1.2 国内外研究现状分析.....	2
1.3 论文的主要研究内容.....	3
1.4 论文的组织结构.....	4
2 围棋直播教学系统的主要技术.....	5
2.1 服务器架构与应用框架.....	5
2.1.1 Nginx 与反向代理.....	5
2.1.2 Django 与 uWSGI.....	6
2.2 流媒体技术与课堂直播.....	8
2.2.1 直播架构.....	8
2.2.2 RTMP 协议与 Nginx-RTMP 模块.....	8
2.2.3 JW Player.....	9
2.3 课堂中的及时互动聊天.....	10
2.3.1 基于 Websocket 实现长连接全双工通信.....	10
2.3.2 Redis 与聊天消息的订阅发布.....	11
2.4 本章小结.....	12
3 基于 RTMP 流媒体技术的围棋直播教学系统的分析.....	13
3.1 围棋直播教学系统的需求描述.....	13
3.1.1 需求背景.....	13
3.1.2 需求说明.....	13
3.2 围棋直播教学系统的需求分析.....	15
3.2.1 围棋直播教学系统的交互模型.....	15
3.2.2 围棋直播教学系统的结构模型.....	22
3.2.3 围棋直播教学系统的行为模型.....	24
3.3 围棋直播教学系统的非功能需求.....	28
3.4 本章小结.....	28
4 基于 RTMP 流媒体技术的围棋直播教学系统的设计.....	29
4.1 围棋直播教学系统的概要设计.....	29
4.1.1 围棋直播教学系统的软件体系结构.....	29
4.1.2 围棋直播教学系统的功能模块结构.....	30
4.2 围棋直播教学系统的详细设计.....	31
4.2.1 基本信息与虚拟钱包.....	31
4.2.2 创办学校与开设课堂.....	32

4.2.3 围棋资源与参与课堂 .....	34
4.2.4 校园管理与课程管理 .....	36
4.3 围棋直播教学系统的关键技术 .....	37
4.3.1 利用 Nginx-RTMP Module 实现教学直播 .....	37
4.3.2 利用 Websocket 与 Redis 实现课堂聊天 .....	39
4.4 围棋直播教学系统的数据库设计 .....	39
4.4.1 围棋直播教学系统的概念数据模型 .....	39
4.4.2 围棋直播教学系统的物理数据模型 .....	40
4.5 本章小结 .....	42
5 基于 RTMP 流媒体技术的围棋直播教学系统的实现与测试 .....	43
5.1 系统开发环境简介 .....	43
5.2 围棋直播教学系统的实现 .....	43
5.2.1 后端服务器配置 .....	43
5.2.2 功能模块的实现 .....	45
5.3 围棋直播教学系统的测试 .....	49
5.3.1 测试环境 .....	49
5.3.2 功能性测试 .....	49
5.3.3 非功能性测试 .....	50
5.4 围棋直播教学系统的运行截图 .....	54
5.5 本章小结 .....	58
6 结论与展望 .....	59
6.1 结论 .....	59
6.2 展望 .....	60
致 谢 .....	61
参考文献 .....	62
学位论文知识产权权属声明 .....	65

## CONTENTS

1	Preface .....	1
1.1	Significance and Background .....	1
1.2	Glance of Current Research Status .....	2
1.3	Main Research Contents of Thesis .....	3
1.4	Organization of Thesis .....	4
2	Main Technology of Online Go Teaching System .....	5
2.1	Server Architecture and Application Framework .....	5
2.1.1	Nginx and Reverse Proxy .....	5
2.1.2	Django and uWSGI .....	6
2.2	Streaming Media Technology and Classroom Live .....	8
2.2.1	Broadcast Architecture .....	8
2.2.2	RTMP and Nginx-RTMP Module .....	8
2.2.3	JW Player .....	9
2.3	Timely Interactive Chat in Class .....	10
2.3.1	Long Connection Full Duplex Communication .....	10
2.3.2	Redis and Subscription to Chat Messages .....	11
2.4	The Summary of This Chapter .....	12
3	Analysis of Go Live Teaching System Based on RTMP Streaming Media .....	13
3.1	Requirements Description of Go Live Teaching System .....	13
3.1.1	Requirements Background .....	13
3.1.2	Requirements Statements .....	13
3.2	Requirements Analysis of Go Live Teaching System .....	15
3.2.1	Interactive Model of Go Live Teaching System .....	15
3.2.2	Structural Model of Go Live Teaching System .....	22
3.2.3	Behavior Model of Go Live Teaching System .....	24
3.3	Non-functional Requirements of System .....	27
3.4	The Summary of This Chapter .....	28
4	Design of Go Live Teaching System Based on RTMP Streaming Media .....	29
4.1	A Summary Design of The Go Live Teaching System .....	29
4.1.1	The Software Architecture of Go Live Teaching System .....	29
4.1.2	The Function Module Structure of Go Live Teaching System .....	30
4.2	Detailed Design of Go Live Teaching System .....	31
4.2.1	Basic Information and Virtual Account .....	31
4.2.2	Establishing Schools and Opening Classes .....	32
4.2.3	Go Resources and Participating in Class .....	34
4.2.4	Management of School and Classroom .....	35
4.3	The key technology design of Go Live Teaching System .....	37

---

4.3.1	Teaching live broadcast based on Nginx-RTMP Module.....	37
4.3.2	Chatroom based on Websocket and Redis.....	38
4.4	Database Design of Go Live Teaching System.....	40
4.4.1	Concept Data Model of Go Live Teaching System.....	40
4.4.2	Physical Ddata Model of Go Live Teaching System.....	42
4.5	The Summary of This Chapter .....	42
5	Implementation and Test of Go Live Teaching System .....	43
5.1	Introduction to System Development Environment.....	43
5.2	The Implementation of Go Live Teaching System.....	43
5.2.1	Backend Server Configuration .....	43
5.2.2	Implementation of Functional Modules .....	45
5.3	Test of The Go Live Teaching System.....	49
5.3.1	The Test Environment.....	49
5.3.2	Functional Test.....	49
5.3.3	Non-Functional Test .....	50
5.4	The Running Screenshot of Go Live Teaching System.....	54
5.5	The Summary of This Chapter.....	58
6	Conclusions and Suggestions .....	59
6.1	Conclusions.....	59
6.2	Suggestions.....	60
	Acknowledgements .....	61
	References .....	62
	Declaration .....	

## 1 绪论

### 1.1 选题意义及应用背景

随着互联网和人工智能的发展，围棋这一有着悠久历史的益智类游戏重新走到大众的关注之下<sup>[1]</sup>，而且相关的话题度也越来越高。但围棋教学培训却形式单一，且质量无法得到保障，由于大部分的围棋教学集中在线下的围棋培训兴趣班之中，优秀的围棋教师、高水平的围棋棋局等围棋资源受到线下的地域性等限制因素，无法得到有效的推广。因此在互联网的背景之下，建立线上的围棋教学平台是一件对于推广围棋学习，普及围棋文化非常有意义的事情<sup>[2]</sup>。

围棋作为中国传统文化的瑰宝，起源于春秋战国时代，在两晋南北朝得到发展，在唐宋时期已经十分流行，甚至影响到日韩<sup>[1]</sup>。明清时期国内也涌现出众多的围棋流派和名手。发展至今，随着互联网和人工智能的日趋成熟，基于人机对弈的围棋成为大众口中热议的话题。尤其是，2016年3月，由Google旗下DeepMind团队开发的人工智能程序AlphaGo与人类围棋世界冠军、职业九段棋手李世石进行的围棋人机大战以及2017年5月，在中国乌镇围棋峰会上，AlphaGo与排名世界第一的世界围棋的冠军柯洁对战，这两则新闻使大众对围棋有了更多的关注和兴趣<sup>[3]</sup>。围棋，作为一项经久不衰而且越来越收到大众，甚至是国家层面关注的有益运动，亟待推广。并且在弈棋的过程，对于棋手的智力、品质、意志、体力等方面的磨练也有诸多好处，尤其是对于处于青少年时期的智力发育，性格培养等都有十分有益<sup>[4]</sup>。因此，国内各地的线下围棋培训机构也越来越多，参与学习围棋的青少年的数量也日益激增，围棋教学的规模也越来越大。可想而知，围棋培训的市场规模也蕴含着巨大的发展空间。

但随着围棋以及围棋学习热度的提升，围棋教学的弊端也日益显现。目前国内外的围棋教学都是以线下培训为主要形式，这种学习形式的短处和痛点包括培训的成本高、围棋教师水平良莠不齐、围棋课程的设置无法满足不同段位的选手等等。因此，作者致力于开发一套线上的围棋教学平台，面向社会大众群体，更加公平有效的分享或传播优秀的围棋资源。线上的围棋直播教学系统<sup>[5]</sup>，不受办学地址限制可以大大节约成本，可使围棋学生更加专注于围棋学习上的投资；系统面向学生，可以公平的参与围棋教学，人人均可学习围棋；系统面向教师，可以将自己的围棋教学能力实现利益最大化；系统划分段位等级，用户可以选择合适自己的段位参与教学或者学习；系统采取收费机制，利用互联网将围棋蕴含的巨大利益效益最大化。

正是基于以上的背景和对围棋教学的思考，开发出一套达到上述目的的围棋线上教学系统可以适应当前围棋运动日益兴盛的局面，也可以更加快速有效的普及围棋知识和围棋文化。并且相比于线下培训，合理的段位选择可使围棋教学活动更加全方位和立体化，面向更多更高段位的围棋选手。完善的购买支付系统也会促进围棋教学的公平性与合理性，有利于围棋推广。

## 1.2 国内外研究现状分析

当前, 围棋运动在世界范围内都呈现日渐兴盛之势。围棋教学却是各国围棋发展的短板。国内外对于推广围棋发展采取的方法, 更多的是在围棋赛事上的投入, 并利用围棋赛事进行盈利, 缺乏真正大规模的推动围棋教学发展的行动。

### 1) 国内围棋教学以及线上系统的发展

近年来, 随着围棋赛事的增多, 国内围棋教学的发展越来越受到大众甚至国家层面的关注。尤其是《2017 年普通高等学校运动训练、武术与民族传统体育专业招生管理办法》<sup>[6]</sup>的出台, 该文件将围棋纳入“单招”项目, 意味着围棋可以作为学生的特长纳入高考成绩, 这对于围棋和围棋教学的发展是重要的鼓励性政策。并且在 2017 年 5 月 28 日下午, 在山东济南举行的中国围棋教学高峰论坛上, 来自教育界、围棋行业、金融界、中科院的专业人士和学者共同探索在互联网+时代围棋文化教育产业的共享发展之路, 围棋教学在国内越来越受到重视。新的时代背景下, 利用互联网来普及围棋文化, 推广围棋教学, 平衡围棋教学资源是一件于国于民都利好的事情<sup>[7]</sup>。而且随着国内围棋爱好者和学习者的增多, 围棋直播教学将有着巨大的经济利益和市场潜力。

然而, 国内的围棋教学却还缺乏科学的教育模式和成熟的教学系统。国内的围棋培训机构由于缺乏有效的市场监管, 导致规模混乱<sup>[1]</sup>。且各地的培训机构主要面向的是幼儿和少年的围棋启蒙教育, 以培养学生对围棋的爱好为主。专业的围棋培训机构也面临着巨大的围棋教师的缺口, 许多教师并非专业的围棋棋手, 或者有些水平较高的围棋老师本身并未接触过教育心理学相关知识, 属于退役的职业选手或水平较高的业余选手。围棋教学不应该局限于围棋线下培训。利用更高的平台降低教育成本, 平衡教育资源, 使二三线城市也可以接收到先进的围棋资源是开发围棋线上教学系统的初衷。

国内的围棋直播教学系统更是发展缓慢。目前, 国内与围棋相关的线上系统主要是各类围棋培训机构的招聘官网, 围棋机构的赛事系统, 或者是提供给业余选手的对弈游戏平台。围棋培训机构的招聘官网和赛事报名系统的目的更多的是服务线下教育, 其系统只提供部分免费的教学视频资源, 对围棋的学习用处不大; 一些围棋对弈游戏上午平台, 可以提供给围棋学习人员课余练习围棋对弈的机会, 但对于围棋教学, 尤其是围棋礼仪, 围棋文化等并没有提供太大帮助。其中, 先恒课堂等网站提供了在线的围棋教学, 但只有个别教师入驻, 缺乏完整的围棋教学体系, 而且并非专业的围棋教学网站; 101 围棋网提供给用户诸多围棋题库, 包括名师棋谱等, 但属于自学范畴。

国内的围棋直播教学系统有待进一步的发展。但国内, “非围棋类的教学系统”近年来发展迅速, 类似网易云课堂、慕课网等提供的直播教学的教育平台值得借鉴。学生可在教师直播过程中, 借助网络与老师实时互动。围棋线上系统应该学习这些教学系统的模式, 借助互联网, 建立符合各段位学生要求的专业围棋教学直播平台, 通过学生与老师的互动, 可以更加公平有效科学的推广围棋教学。

### 2) 国外围棋教学以及线上系统的发展

近年来, 世界范围内围棋发展和围棋教学的热点主要集中在中, 日, 韩三国。相比

欧美国家,围棋在中日韩的普及程度更高,围棋的发展和围棋人才的培养也更加成熟。欧美国家的围棋培养主要是围棋爱好者的自学和参加围棋俱乐部<sup>[8]</sup>,其围棋教学尚未形成规模,围棋直播教学系统更是几乎没有发展。中日韩三国的围棋培养也各有特色。

韩国的围棋教学,主要是通过“放学后教育”为主,以学生放学后的兴趣培训班为单位进行围棋学习。这种方式只是针对处于启蒙阶段的中少年,对于围棋专业或职业水平的培养缺乏有力的支持。在韩国还有一种围棋教学形式是围棋教室,类似中国的围棋培训机构,依然受到地域性限制,不利于围棋学习的广泛开展。而与围棋相关的线上新网站大多数同样是围棋比赛或者围棋新闻,与围棋教学相关的线上系统寥寥无几。

相比于韩国,日本围棋的发展更加成熟。这源于近现代以来围棋运动在日本十分受欢迎。在日本,围棋很早已经是中少年课外小组的主要活动之一,在高中阶段,甚至围棋已经是必修课程。而在专业围棋选手的培训上,以围棋道场形式的培训机构采用内弟子制度也培养出许多一流的围棋高手,对围棋的专业选手培养十分有利。然而,相比于线上教学系统,通过网络无需有具体的师承关系,业余选手也可以参与高水平的围棋学习。同样,高水平的围棋教师可以面向所有学生传授知识,这对于围棋的发展更加有好处。日本的围棋直播教学系统也是以围棋游戏的形式,在网络上与其他棋手一对一的对弈,并没有形成具有从启蒙阶段到高水平围棋棋手的专业在线教学系统。

围棋在世界范围内的主要学习以培训为主,在线教学系统目前缺乏成熟的案例。但同国内一样的是,针对类似编程等技能的教学平台却有很多成熟的案例。类似 Coursera、Skillshare 等都值得借鉴,以类似的方式传播围棋的教育教学,和理公平的利用互联网分配围棋资源对于围棋的发展大有裨益。

### 1.3 论文的主要研究内容

本论文的目标是开发一款线上的围棋直播教学系统。该系统面向围棋学生和围棋教师,两类用户都维护了基本的信息,都拥有唯一的虚拟钱包。虚拟钱包是用户接收系统服务所必需的。围棋直播教学系统为教师用户提供创办学校和开设课堂权限,学生通过购买进入相应课堂参与围棋学习。本论文具体的研究内容如下:

1) 用户身份信息维护。围棋直播教学系统应区分两类用户,学生和教师。学生用户通过系统学习围棋知识,教师用户通过系统推广围棋教学。为保证教学系统的围棋教学高效的展开,教师身份需系统审核。系统为两类用户均提供注册、登陆、基本信息查看与修改的等功能。其中,教师用户不可直接注册,须有学生用户通过身份转换模块进行升级转变。

2) 虚拟钱包管理。围棋直播教学系统内为每个用户维护了虚拟的钱包,用户可对虚拟货币“棋子”进行充值可提现。虚拟货币支撑了围棋直播教学系统的购买支付功能,学生用户参与课堂,教师用户开设学校和课堂均需要进行购买花费。

3) 创办学校与开设课堂。由于只有教师用户拥有建校和开课权限,普通学生用户需通过升级标签转为教师用户。创办学校与课堂可使同一个围棋教师拥有多个围棋学

校或直播课堂，课堂之间互相分离，同一学校内部又可以管理所有课堂统一的资源，更加高效的服务于围棋教学活动。

4) 围棋资源与参与课堂。围棋资源与参与课堂模块包含自学与加入课程学习两种方式。系统提供各段位的围棋视频，用户可自主选择观看，学生用户对学校和课堂内分享的课件等围棋资源也拥有查阅权限。学生用户参与课堂过程中，教师用户采用直播形式分享围棋知识，学生用户可实时观看围棋教师的围棋走势和讲解，观看过程中，系统为教师用户和学生用户还提供了实时交流模块，方便教学活动展开。

5) 校园管理与课程管理。学生用户对于加入的校园和课堂拥有查看对应信息的权限，例如学生用户需要随时查看开课时间，学生用户对于不喜欢的课堂可以删除。教师用户对于所开设的学校和课堂拥有更加高级的权限，更新校园信息和课堂信息，删除开设的学校与课堂，除此之外，系统还提供校园内的文件管理，课堂中的学生管理等。

## 1.4 论文的组织结构

论文从围棋直播教学的实际需求出发，以软件工程的方法为指导撰写各章节内容，依照系统的分析、设计，实现和测试等开发流程来完成论文组织结构的编写工作。本论文的组织结构如下：

第 1 章 绪论。主要介绍论文的研究背景和开发围棋直播教学系统的意义，对国内外的发展现状做了分析，介绍了论文的主要工作内容和论文的组织结构。

第 2 章 围棋直播教学系统的主要技术。介绍开发围棋直播教学系统过程中所使用的系统框架以及主要技术，包括实现整个系统所采用的 MTV 模式，与课堂直播相关的推流拉流以及 Nginx-RTMP 模块，以及实现课堂内聊天室的 websocket 和 Redis。

第 3 章 围棋直播教学系统的需求分析。对围棋直播教学系统进行需求描述，分析完成围棋教学所需要实现的各项功能。在分析系统需求的过程中，分别以学生用户和教师用户的角度，通过功能模型，结构模型和行为模型对系统所需要的服务以及用户与系统的交互行为进行分析。

第 4 章 围棋直播教学系统的设计。通过需求阶段的分析，完成了系统的概要设计和详细设计，用总体架构图说明系统后端的架构设计，用功能模块图表示系统的各模块及其子模块之间的关系，并通过类图说明各模块内部的类间关系，通过时序图说明两类用户接受系统服务时，模块内部的类间交互。最后依据各模块所需要维护的用户与课堂等信息，完成了数据库的设计。

第 5 章 系统的实现与测试。本章完成了系统的各个模块。通过系统配置和模块内代码实现的形式论述了系统的各模块以及模块之间的交互与关联。最后，完成了系统的功能性和非功能性的测试，通过测试用例说明系统是否符合需求，性能是否达标。

第 6 章 结论与展望。本章对本文的研究内容和研究成果进行总结，归纳出论文和系统的不足之处，并对下一步的研究工作做出展望。



## 2 围棋直播教学系统的主要技术

本章主要介绍围棋直播教学系统后端采用的服务器架构以及系统实现过程中所需要的主要技术。包括整体系统设计所采用的基于 MTV 模式的 Django 框架，负责反向代理以及处理直播推流和拉流的 Nginx 与 Nginx-RTMP 模块，负责教学互动所采用的 websocket 技术与 Redis 存储系统,为后文的实现与测试环节做技术支撑。

### 2.1 服务器架构与应用框架

#### 2.1.1 Nginx 与反向代理

Nginx 由俄罗斯的开发人员 Igor Sysoev 于 2005 年编写的一个免费、开源、高性能的 HTTP 服务器，也可提供高效的反向代理服务<sup>[9]</sup>。Nginx 正因为其工作稳定，功能集丰富，配置简单，资源占用低，并发能力强而赢得了广阔的市场，例如国外的 Github, Facebook, WordPress 等，国内的迅雷，网易，新浪等均采用 Nginx 为其提供服务。

本文中实现的系统采用 Nginx 作为反向代理服务器，并利用 Nginx 处理静态文件。Nginx 在作为反向代理服务器和处理类似 css, javascript 等静态文件文件方面有独特的优势。如图 2-1 所示，反向代理（Reverse Proxy）方式是指以代理服务器来接受网络上的请求，然后将请求转发给内部网络上的应用服务器，并将从应用服务器上得到的结果返回给网络上请求连接的客户端，此时代理服务器对外就表现为一个反向代理服务器。围棋直播教学系统中，Nginx 作为反向代理服务器，将用户动态的业务逻辑请求转发给后台的 uWSGI 服务器,由部署在 uWSGI 应用服务器中的 Django 应用进行处理，返还结果处理到 Nginx, Nginx 转发给用户。

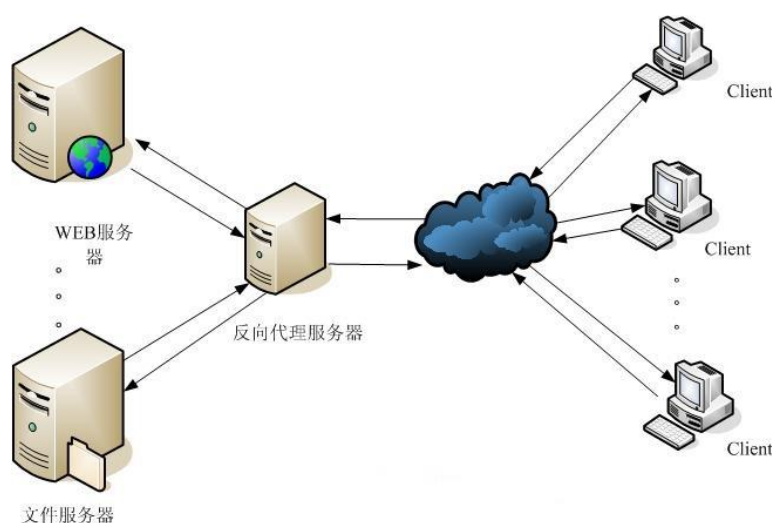


图 2-1 反向代理服务器工作示意图

Nginx 由于本身就是由 C 语言完成的 HTTP 服务器，其提供基本的静态文件访问功能因此，利用 Nginx 本身高效的静态文件处理功能，实现后端静态资源和动态资源

的分离,减轻服务器的压力十分必要。在搭建围棋直播教学教学系统的过程中,本系统利用 Nginx 作为处理 css,javascript,图片等静态文件的服务器。

高度模块化的设计是 Nginx 架构的基础,也带来如下特点:高度抽象的模块接口;灵活性;配置模块的设计使 Nginx 提供了高可配置性、高可扩展性、高可定制性、高可伸缩性;核心模块接口简单化;多层次、多类别的模块设计。Nginx 除了本身提供的核心模块和邮件模块外,允许将第三方模块编译进 Nginx,这大大的方便了 Nginx 的性能扩展,围棋线上教学系统的直播功能就是将 Nginx-RTMP 模块重新编译进 Nginx 来实现的。

### 2.1.2 Django 与 uWSGI

#### 1) Django 开发框架

Django 是一个由 Python 写成开放源代码的 Web 应用框架。是一个基于 MVC 构造的框架<sup>[10]</sup>。但是在 Django 中,控制器接受用户输入的部分由框架自行处理,所以 Django 里更关注的是模型 (Model)、模板(Template)和视图 (Views),称为 MTV 模式<sup>[11]</sup>。它们各自的职责如表 2-1 所示。

表 2-1 MTV 职责

层次	职责
模型 (Model), 即数据存取层	处理与数据相关的所有事务, 包括数据字段的定义以及 与数据库的交互
视图 (View), 即业务逻辑层	处理请求逻辑, 是模型和模板的桥梁
模板(Template), 即表现层	根据视图返回的内容处理页面显示

从表 2-1 可以看出 Django 的模型层负责处理与数据相关的事物, 包括数据的存储, 数据有效性的验证等。通过 Django 的 ORM (对象关系映射) 机制<sup>[12]</sup>, 可以方便开发人员直接利用 python 对象直接操纵数据库, 实现对应数据表项的增删改查。原本需要写原始的 SQL 语句来操作数据库, 通过 Django 只需要调用相应的 API 就能完成一系列操作<sup>[13]</sup>。

视图层用于处理业务逻辑, 仅处理请求和用户提交的数据并决定要返回哪些数据给模板层。通过视图层, 开发人员调用模型层和模板层的相关函数, 实现对用户信息的更新查看与反馈结果的展示。

Django 模板层决定如何通过模板标签展现对应视图层返回的数据, 类似于 MVC 中的表现层。Django 将 MVC 中的视图进一步分解为 Django 视图 和 Django 模板两个部分, 分别决定“展现哪些数据”和“如何展现”, 使得 Django 的模板可以根据需要随时替换, 而不仅仅限制于内置的模板。

Django 框架的 URLconf 机制是使用正则表达式匹配 URL, 然后调用视图中的 Python 函数。这意味着开发者可以搭配自己意愿的 URL 风格。所以, Django 的 MTV 比 MVC 框架考虑的问题要更多。MVC 中, 程序员需要写控制层而在 MTV 模式中, 控制层的东西由 Django 的框架自动完成。而在 Django 中这些底层的工作由

Django 自行处理，开发者无需关注，同时自然减少了代码量，提高了工作效率。

Django 的 MTV 交互以及 URLconf 机制如图 2-2 协同完成用户请求的接受、处理与展示。

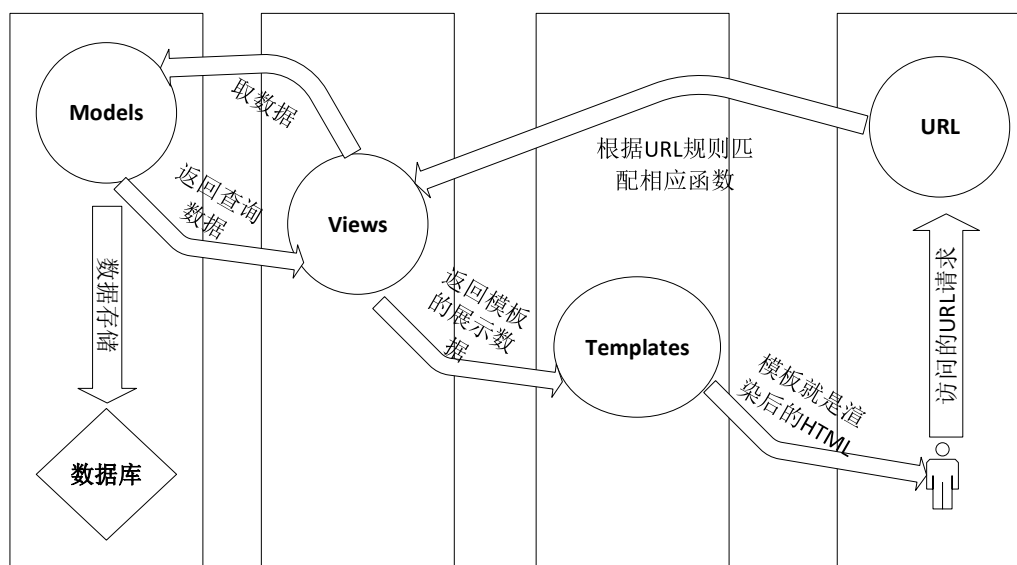


图 2-2 Django WEB 框架

在围棋线上教学系统中，本文利用 Django 的 MTV 模式分别在 Model 层完成学生用户信息，教师用户信息，学校和课程信息等数据表的建立，表段数据的填充以及数据的持久化。并利用 Django 的 ORM 映射，将数据库的表段数据映射到 python 对象，在 View 层操纵 python 数据对象而不是直接操纵数据库。在 View 层，主要处理业务逻辑，包括用户注册登陆，观看直播，即时聊天，上传下载文件以及学校和课程管理在内的逻辑任务在 view 层中有相应的 python 函数负责其逻辑处理。View 层处理完逻辑任务后，将结果以字典，列表等形式返回到 Template 层，Template 层可以像 JSP 一样将 View 层返回的数据以可视化的形式展示给用户。

## 2) uWSGI 服务器

Django 内部提供一个开发服务器，runserver，但由于这个开发服务器没有经过安全测试，而且使用的是 Python 自带的 simple HTTPServer 创建的，在安全性和效率上都是无法达到教学系统的需求的。并且 Django 框架自带的服务器属于 WSGI(the Python Web Server Gateway Interface)接口，目前 python 应用框架的主流服务器以 uWSGI 协议为主。

uWSGI 协议是一个 uWSGI 服务器自有的协议，它用于定义传输信息的类型 (type of information)，每一个 uWSGI packet 前 4byte 为传输信息类型描述，与 WSGI 相比是两种不同的协议，而且 uWSGI 协议具有比 WSGI 更高的传输效率。uWSGI 是一 Web 服务器，它实现了 WSGI 协议、uWSGI、http 等协议。将 Django 部署在 uWSGI 服务器上，可以达到更高的并发量。且由于 uWSGI 服务器采用 master-worker 的进程管理模型，可以更加方便的操纵和管理进程。且 Nginx 中通过 HttpUWSGI Module 与 uWSGI 服务器进行信息交换。

在围棋线上教学系统中，利用 uWSGI 服务器提高请求处理的效率，达到更高的并发访问数量；利用 Django 方便的架构业务应用，快速便捷的处理业务逻辑；Nginx 在系统中负责用户请求的反向代理以及静态文件的返回。

## 2.2 流媒体技术与课堂直播

### 2.2.1 直播架构

流媒体指将音频、视频等多媒体形式的文件以数据流方式在网络中进行传送<sup>[14]</sup>。相对于将这些文件下载后再观看的播放形式而言，流媒体的典型特征是把连续的音频和视频信息压缩后存储到流媒体服务器上，用户可以一边下载一边观看，而不必等待整个文件下载完毕<sup>[15]</sup>。正是由于流媒体技术的这种优越性，该技术广泛应用于视频点播、视频会议、远程教育、远程医疗和在线直播等系统中。流媒体服务器是流媒体应用的核心系统，是运营商向用户提供视频服务的关键平台。流媒体服务器的主要功能是对流媒体内容进行采集、缓存、调度和传输播放。流媒体应用系统的主要性能体现都取决于媒体服务器的性能和服务质量。因此，流媒体服务器是流媒体应用系统的基础，也是最主要的组成部分。在围棋直播教学系统中，主要利用流媒体技术中的视频点播实现免费视频的播放，利用直播实现教学课堂上的实时讲解<sup>[16]</sup>。

在课堂直播中，需要实现教师用户推流，流媒体服务器存储和分发以及学生用户通过浏览器拉流的共同配合实现实时教学。如图 2-3 所示，教师用户可通过 ffmpeg 等推流软件将采集的 PC 桌面视频流或者通过 ffmpeg 将采集的摄像头视频流推送到流媒体服务器<sup>[17]</sup>，流媒体服务器进行缓存和处理，学生可通过访问对应的网页，网页中的 javascript 程序会到指定的流媒体服务器地址进行视频流的拉取，实时观看教师的讲解。围棋线上教学系统中主要的推流与拉流阶段使用到 RTMP 协议，具体需要通过 Nginx 的 RTMP 模块来实现。



图 2-3 直播过程中的推流拉流示意图

### 2.2.2 RTMP 协议与 Nginx-RTMP 模块

RTMP(实时消息传送协议)是 Real Time Message Protocol 的缩写,最初是由 Macromedia 开发的专有协议,用于通过网络在 Flash 播放器和服务器之间传输音频,视频和数据<sup>[18]</sup>。Macromedia 现在归 Adobe 所有,RTMP 也已经发展为一个协议族,包括 RTMP 基本协议/RTMPT/RTMPS/RTMPE 等多种变种。RTMP 工作在 TCP 之上,默认使用端口 1935; RTMPE 在 RTMP 的基础上增加了加密功能; RTMPT 封装在 HTTP 请求之上,可穿透防火墙; RTMPS 类似 RTMPT,增加了 TLS/SSL 的安全功能<sup>[19]</sup>。近

年来,在国内非常火的直播平台,例如斗鱼,虎牙等系统的推流地址都是基于 RTMP 协议。

RTMP 协议属于应用层协议,需要依靠底层可靠的传输层协议(通常是 TCP)来保证信息传输的可靠性<sup>[20]</sup>。在基于传输层协议完成三次“握手”之后,RTMP 协议也要客户端和服务端通过“握手”来建立基于传输层链接之上的 RTMP Connection 链接,在 Connection 链接上会传输一些控制信息,如 SetChunkSize,SetACKWindowSize。其中 CreateStream 命令会创建一个 Stream 链接,用于传输具体的音视频数据和控制这些信息传输的命令信息。RTMP 协议传输时会对数据做自己的格式化,这种格式的消息本文称之为 RTMP Message,而实际传输的时候为了更好地实现多路复用、分包和信息的公平性,发送端会把 Message 划分为带有 Message ID 的 Chunk,每个 Chunk 可能是一个单独的 Message,也可能是 Message 的一部分,在接受端会根据 chunk 中包含的 data 的长度, message id 和 message 的长度把 chunk 还原成完整的 Message,从而实现信息的收发<sup>[21]</sup>。

Nginx 由于支持模块设计,具有非常好的扩展性,无论是 Nginx 官方的模块还是第三方的模块,只需要编译进 Nginx,即可利用 Nginx 的高并发等特性扩展自己的服务。Nginx-RTMP 模块<sup>[23]</sup>是官方推出的使 Nginx 提供流媒体服务的模块,编译进 Nginx 之后,只需要在 Nginx 配置中指定相应的参数,就可以方便的搭建其流媒体服务器。围棋在线教学系统中,利用 Nginx-RTMP 模块搭建流媒体服务器,实现了课堂直播与缓存,以及免费视频播放的功能。再课堂直播中,教师用户需借助第三方软件推流到指定的 RTMP 地址,例如可用 ffmpeg 采集摄像头信息并推流或通过 ffmpeg 软件实现 PC 桌面视频流的推送,在学生端通过网页中的 JW Player 组件可以实现到对应的 RTMP 地址拉流,实现实时课堂直播功能。

### 2.2.3 JW Player

JW Player 是一种基于 flash 的交互式网页媒体播放器<sup>[24]</sup>。它是由 Jeroen 和 Wijering 共同建立的 LongTail Video 所开发,问世于 2005 年。JW Player 包含: FLV Player、WMV Player、Image Rotator 和 Desktop Player 四种,除了最后一种实为桌面播放器,其他三种都是网页播放器。JW Player 作为当今最流行的开源 Flash 网页播放器,可播放 Adobe Flash Player 所支持的媒体,具体包括: FLV、MP4、MP3、AAC、JPG、PNG 和 GIF 等,还支持 RTMP、HTTP、实时视频流、各种播放清单格式、灵活的设置和广泛的 javascript API。此外它还提供多种外观、功能性插件来扩展播放器,以便本文可以分享、推荐、搜索、分析甚至广告投放。Youtube 也层使用 JW Player 作为播放器,国内的慕课网也同样选择 JW Player 作为网页播放器。

围棋直播教学系统采用 JW Player 作为浏览器插件实现对 RTMP 协议地址的拉流,有以下好处:支持所有主流的浏览器;能与播放器交互,播放器必须提供必要的 api;可以定制外观,方便后期拓展;支持 flv、mp3、mp4 以及 RTMP 格式;有详尽的开发文档。

## 2.3 课堂中的及时互动聊天

### 2.3.1 基于 WebSocket 实现长连接全双工通信

WebSocket 协议是一种基于 TCP 并且 HTML5 规范中被引用的新型网络协议<sup>[25]</sup>。WebSocket 与 HTTP 是两种不同的协议，这两个协议都位于 OSI 模型中的第 7 层，且第 4 层的 TCP。在 WebSocket 出现之前，很多网站都是通过轮询(polling) 来实现即时通讯。轮询是指在特定的时间间隔（如每 1 秒），由浏览器对服务器发出 HTTP 请求，然后由服务器返回最新的数据给客服端的浏览器，这种方式有一个很大的弊端，就是会占用很多的带宽<sup>[26]</sup>。最新的轮询效果是 Comet – 用了 AJAX。但这种技术虽然可达到全双工通信，但依然需要发出请求。而使用 WebSocket，浏览器和服务器只需要要做一个握手的动作，然后，浏览器和服务器之间就形成了一条快速通道，两者之间就直接可以数据互相传送<sup>[27]</sup>。而且它为本文实现即时服务带来了两大好处：节省资源，互相沟通的 Header 是很小的-大概只有 2 Bytes；推送信息，不需要客户端请求，服务器可以主动传送数据给客户端。websocket 这通信是通过 TCP 端口号 80（或在 TLS 加密连接的情况下为 443）完成的，这对于那些使用防火墙阻止非 web 互联网连接的环境是有益的。在 Firefox,Chrome 等常用的浏览器中，均以及实现了 websocket 协议。

实现 websocket 连线过程中，需要通过浏览器发出 websocket 连线请求，然后服务器发出回应，这个过程通常称为“握手”。“握手”过程中，客户端的请求头和服务器的相应头如图 2-4 所示。

```

客户端请求头:
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
Origin: http://example.com

服务器响应头:
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: H5mrc0sM1YUkAGmn50PpG2HaGWk=
Sec-WebSocket-Protocol: chat

```

图 2-4 websocket 的请求头和响应头

在 WebSocket 对象的构造函数 `var ws = new WebSocket(url='ws://ip:port',protocols)` 中有两个参数，其中 `protocols` 是可选参数，用于指定连接子协议,如 `ws`, `wss` 等；而 `url` 参数是必须的，它指定了要连接服务器端的地址及端口号。连接过程的状态有 `Connecting`、`Open`、`Closing` 和 `Closed` 四种状态<sup>[28]</sup>，它们都保存在 `readyState` 属性中，分别代表了正在连接状态、已连接状态、正在关闭状态和连接已关闭状态。通过调用 `Websockets` 对象的 `send()` 方法可以向服务器传输文本或二进制的的数据。调用 `close()` 方

法会在客户端发出关闭报文主动关闭双向连接，从而触发连接关闭事件。在接口的定义中还有相应的事件处理器如 `onopen()`, `onmessage()` 等来响应服务器的事件。客户端的 Websocket 对象共有以下四个事件用于响应服务器：

```
onopen(): 连接建立时触发;
socket.onopen = function(event) { //连接建立处理函数 }
onmessage(): 收到消息时触发;
socket.onopen = function(event) { //消息处理的函数 }
onerror(): 连接出现异常时触发;
socket.onopen = function(event) { //对错误的处理 }
onclose(): 关闭连接后触发;
socket.onopen = function(event) { //连接关闭后的动作 }
```

由于浏览器端对协议提供了很好的支持，所以开发人员在客户端进行相应开发时只需了解如何调用即可，无需关注协议的具体实现过程。

Django 中提供了 `dwebsocket` 模块使 Django 项目方便快捷的搭建支持 websocket 通信的应用。`dwebsocket` 需要协程的支持，所以在围棋线上教学系统中，应安装 `eventlet` 或 `greentlet` 等模块。在 Django 视图中，只需要在处理 websocket 的函数前使用 `@qccpt_websocket` 装饰器即可。

### 2.3.2 Redis 与聊天消息的订阅发布

Redis(Remote DIctionary Server)是一个由 Salvatore Sanfilippo 写的 key-value 存储系统<sup>[29]</sup>。Redis 是一个开源的使用 ANSI C 语言编写、遵守 BSD 协议、支持网络、可基于内存亦可持久化的日志型、Key-Value 数据库，并提供多种语言的 API。它通常被称为数据结构服务器，因为值（value）可以是 字符串(String)，哈希(Map)，列表(list)，集合(sets) 和 有序集合(sorted sets)等类型。由于 Redis 使内存数据库，数据的读写速度很快。

Redis 提供了发布订阅功能，可以用于消息的传输，Redis 的发布订阅机制包括三个部分，发布者，订阅者和 Channel。如图 2-5，发布者和订阅者都是 Redis 客户端，Channel 则为 Redis 服务器端，发布者将消息发送到某个的频道，订阅了这个频道的订阅者就能接收到这条消息。Redis 的这种发布订阅机制与基于主题的发布订阅类似，Channel 相当于主题。

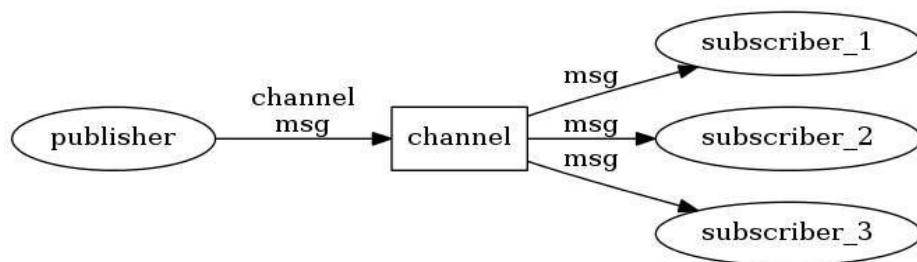


图 2-5 Redis 的订阅和发布

在围棋直播教学系统中，即时聊天通过不同的学校和课堂名称作为发布和订阅的 Channel，可实现不同课堂对应不同聊天室的划分，学生用户和教师用户均可向 Redis 发布消息，同一课堂的其他学生自动订阅消息，实现了消息的互通。

## 2.4 本章小结

本章主要介绍了实现围棋直播教学系统所需要的主要技术，2.1 节介绍了系统所采用的整体后端架构和开发模式，2.2 节讲述了实现课堂教育所采用的直播所采用的技术和方法，2.3 节介绍的技术是分开不同课堂实现即时互动聊天的主要技术。依据这些技术，系统在开发中分别对应不同模块来实现系统的需求。



### 3 基于 RTMP 流媒体技术的围棋直播教学系统的分析

上一章对开发本系统所涉及的主要技术进行了介绍，接下来将对围棋直播教学系统进行需求分析。本章将以围棋教学面临的问题为切入点描述系统的业务需求，说明系统应当提供的教学服务，阐述服务过程中的用户角色。本章将根据不同的用户类别建立系统的功能模型，结构模型和行为模型，并给出了详细的分析和说明。

#### 3.1 围棋直播教学系统的需求描述

##### 3.1.1 需求背景

随着围棋运动越来越受到人们的喜爱，通过进入线下培训机构去学习围棋的人也越来越多。但由于线下培训机构的地域性限制以及培训价格的不透明性，围棋的推广和围棋的大众化学习进程都受到很大阻碍。

经过市场调研，可以发现，线下培训机构多是面向幼儿或者是中少年的学生，且围棋教学以入门培训为主。类似西安地区的奕学园、合肥市方圆围棋培训学校等，都是针对围棋初学者的基础教育。且这类学校的线上系统十分简陋，只有简单的学校介绍和报名渠道，主要的围棋教学还是以线下教学为主。而北京快乐围棋培训中心、聂卫平围棋道场等培训机构，虽有针对较高段位水准的课程，但受限与地域性因素，且无良好的网络课程，使围棋资源无法在更大范围内分享和利用。可见，线下的围棋培训学校对于围棋“教”“学”活动的推广十分有限，而且无法将围棋学习的市场规模激发出来。

综上，围棋直播教学系统的初衷采用互联网的手段最大化利用围棋资源，并且利用直播技术保证围棋教学过程真实性，使围棋课堂拥有线下课堂的亲切感<sup>[30]</sup>。围棋直播教学系统不仅面向围棋学生，而且应当为线上培训机构提供通道。鼓励线下的培训机构，采用本系统建立网上学校，统一管理校内课程和教学资源。且系统内课程收费采用明码标价机制，利于围棋培训市场的公平竞争。

##### 3.1.2 需求说明

围棋直播教学系统，要解决的问题是将包括教师和教学技巧在内的围棋教学资源进行整合，并重新利用互联网进行分配，使围棋教学不受地域、时间、空间的影响，同时为数量庞大的围棋学子提供一套可以覆盖基础到高级别段位的学习系统，并且学生可以自由选择合适的教学风格。围棋直播教学系统的另一特色是围棋学校的创办以及校内课程的开通都采取统一收费机制，学生用户参与直播课堂、升级为教师用户、以及教师用户创建学校、开设课堂需要缴纳费用，教师用户利用课堂实现盈利。围棋直播教学系统采用的虚拟货币交易系统可以有效避免收费混乱，学习成本高昂的问题。公开的围棋课程价格也有利于促进围棋教学的良性竞争。

针对围棋直播教学系统所要解决的问题，可将系统的需求划分为以下几个部分：

### 1) 用户基本信息管理需求描述

系统面向学生和教师用户进行围棋教学，通过线上教学的方式解除线下的诸多弊端，但不同身份用户的基本信息管理是最基础，也是最为必要的。系统需要获取用户的账户以及 ID 等作为用户唯一身份的标识，同时用户可以增加、更改、删除自己的基本信息，例如学生的出生年月、性别，教师用户的建校年份和办学宗旨等。系统灵活的存储和管理基本信息，为用户提供学习保障。

### 2) 虚拟货币账户管理需求描述

盈利同样是围棋直播教学系统的核心目标之一，因此系统面向教师用户和学生用户采取收费建校、收费开课、付费学习的方式。系统中为用户维护了虚拟货币的账户，用户可以在钱包中查看自己的账户余额，在系统中需要收费之处需要花费虚拟货币。同样，用户在盈利之后也可以进行提现，在余额不足时进行充值。

### 3) 创办学校与开设课堂需求描述

围棋直播教学系统面向不同的用户身份应授予不同的权限。为避免教学课堂的混乱，普通学生用户只拥有参与课堂的权限；为避免围棋学校的良莠不齐，保证系统中的课堂教学质量，学生用户需要升级为教师用户才可拥有创办学校的权限。转换为教师用户应当提供相关围棋水平或教学资质的证明，系统审核通过后升级成功。教师用户拥有创立学校和开通课堂的资格，学校数量和课程数量不受限制，课堂应在对应学校名下，属于学校管理。教师用户在创办学校时，需要提交学校所属段位、学校办学宗旨等相关信息，系统进行维护；教师用户开设新课堂时，需要提供课堂所属段位、课堂名称等资料，开设课堂成功时，系统为教师用户提供推流地址。教师在约定时间进行直播，学生在约定时间参与课堂即可；学生用户的升级以及教师用户创办学校、开设新的课堂均需要向系统缴纳一定的费用，完成支付后方可拥有对应的资格。

### 4) 围棋资源与参与课堂需求描述

围棋直播教学系统最主要的目标是为线上不同身份的用户提供统一且灵活的课堂，学生可以在线上课堂中最大化自己的学习效果是系统最为主要的需求。通过直播的形式，系统可使学生和教师做到类似线下课堂的教学形式，直播的画面生动的展示了围棋教师的教学风采，学生对于知识的接受也可以更加亲切和客观。直播教学过程中，学生和老师通过聊天室可以实时沟通，教师按教学情况调整课堂内容，学生通过聊天室的提问教师可以当场回答，教师提出的问题学生之间可以互相讨论学习。

课堂中的课件和学生后期的作业，都可以通过系统进行上传下载，为学生的课前预习和课后复习提供了一定的保障。参与课程的学生默认加入该课程所属的学校，对学校主页的共享资料也可随时下载查看。除学校与课堂中，教师用户提供的共享围棋资源以外，系统还为所有用户提供已分类的围棋视频资源，用户可在选择段位后查看相应视频。

### 5) 校园管理与课程管理需求描述

完善的教学系统应该使教学课程彼此分离，教学资料可以彼此共享。学校和堂对

参与其中的学生应有管理权限。围棋直播教学系统中，学生与教师均可参与课堂听课学习，参与的课堂可以有多项。对于参与的课堂，用户应当拥有校园与课堂信息查看，校园与课堂删除的权限，删除后对相应的学校和课程内分享的学习资料无查阅权限。

教师用户对自己开设的所有学校和课堂默认拥有参与权限。除此之外，教师用户应当拥有所开设学校与课堂的最高权限，包括添加新的学校和课堂，更新对应的信息，删除发布的学校和课堂等。其中，教师用户删除学校，则该学校下所有课程均删除；教师用户删除课堂，则参与该课堂的学生课程表中也会删除对应项目。学校和课堂对参与其中的学生用户拥有管理权限，可以统计在校人数、查看课堂参与人数等基本情况。学校主页应提供共享文件，供所有课程共用。课堂主页提供学生管理功能，可删除课堂内的破坏分子。

## 3.2 围棋直播教学系统的需求分析

本节首先分析系统的功能性需求，利用用例图和用例描述为系统建立交互模型，划分系统的需求；然后通过类图描述系统的内部信息结构以及彼此间交互行为，建立系统的结构模型；最后通过活动图对系统中的动态行为进行描述，说明人与系统的交互操作，建立系统的行为模型。

### 3.2.1 围棋直播教学系统的交互模型

#### 1) 系统用例分析

这一小节主要描述基于 RTMP 流媒体技术的围棋直播教学系统的交互模型，为了更清楚的表达系统的功能性需求，本节采用 UML 中的用例图来阐述系统的总体功能需求划分。之后，采用用例图和用例描述表结合的方式说明各具体的功能需求所应提供的服务。

围棋直播教学系统的首要目的是提供给围棋学子一个学习围棋的环境。在系统中，使用者可以通过参与围棋课堂直接观看教学直播，下载教学课件，参与课堂问答和讨论，这类使用者属于学生用户。系统提供了用户升级和创办学校、开通课堂的功能，用户可以根据自身资历创办不同段位水准的学校和课堂，在开通课堂后可以面向学生收费，这类用户属于教师用户。并且由于学生在用户升级为教师用户过程中，需要系统管理员审核相关资历，因此，系统管理员也属于系统使用者角色，系统管理员也承担了用户信息维护、学校及课程的信息以及其中文件数据维护、用户课程表管理等作用。由此可知，围棋直播教学系统将用户定义为三种角色，分别为学生、教师、系统管理员。其中系统管理员角色不对普通用户开放，由系统的后端操作人员执行其功能，后台管理员将负责后台各类数据的维护。

针对需求描述中的主要需求，可将围棋直播教学系统通过基本信息与虚拟钱包、创办学校与开设课堂、围棋资源与参与课堂、校园管理与课程管理四个大的服务功能完善系统要求。四个服务功能均需要学生、教师、后台管理员三类用户角色参与。通过如上的需求描述和系统角色的分析，可划分出系统总体用例图如图 3-1 所示。

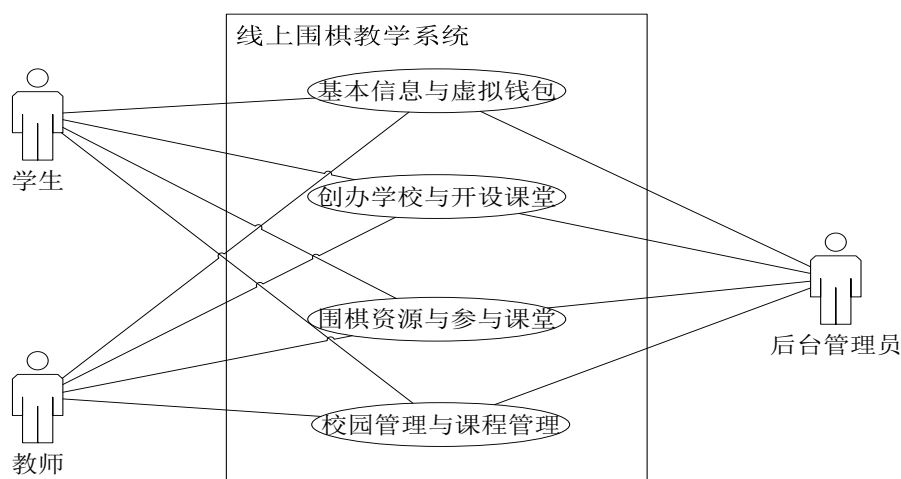


图 3-1 系统总体用例图

基本信息与虚拟钱包中，学生和教师用户的基本信息数据和唯一身份标识以及用户的唯一账户数据，都需要系统提供基础的管理功能。用户身份的唯一性是系统提供公平稳定服务的前提。虚拟钱包为系统添加了货币流通功能，为后续具体的直播以及教学服务奠定了基础。系统管理员负责维护和管理各类信息。创办学校与开设课堂服务功能中，教师用户可以借助系统开设网上学校，在学校内部分享包括课件、校园通知等各类文件；教师用户可以在该学校内开设收费课堂，对学生收费后进行围棋教学，从而实现围棋知识的传播以及盈利。学校提供了资源的统一管理，课堂提供了学生接受具体知识的隔离性，保障了教学活动的高效展开。系统管理员负责资料审核，检查学校和课堂的开设资格，并维护学校与课堂数据。围棋资源与参与课堂中的围棋资源包括三类，系统提供的免费视频资料，学校内的文件资料，以及参与课堂内的教师课件和学生作业。参与课堂是学生用户学习围棋知识的主要途径，也是围棋直播教学系统的初衷。通过参与课堂的形式，学生应当可以直接观看教师的实时直播教学，与教师实时沟通交流，更加直观的学习围棋知识使围棋学习更加活泼生动。校园管理与课程管理模块，系统将维护学生的课程表数据，教师创建的学校和课堂信息。为学生管理自己的学习活动，教师管理自己的教学活动提供了基础服务。除此之外，学校内的文件管理以及课堂内参与学生的管理，也对围棋教学活动的展开提供了很大的便利。在此后的章节中，将详细描述各功能需求部分的具体要求。

## 2) 基本信息与虚拟钱包用例分析

用户基本信息与虚拟钱包管理是围棋直播教学系统的核心需求之一，主要用户角色包括学生用户和教师用户，后台管理员负责维护用户的基本数据。如图 3-2 所示为基本信息与虚拟钱包用例图，其中包括了用户的基本信息管理和虚拟钱包管理两大部分。

基本信息管理面向用户提供了用户信息注册，用户信息查看，用户信息修改三部分。用户使用围棋直播教学系统前，必须进入指定的网页进行注册，拥有唯一账号后方可登陆系统使用系统提供的服务；用户在使用系统过程中，应当可以随时查看和更改自己的基本信息，包括年龄、性别、昵称等，唯一的账户 ID 是用户身份唯一性的表示，

不可更改。

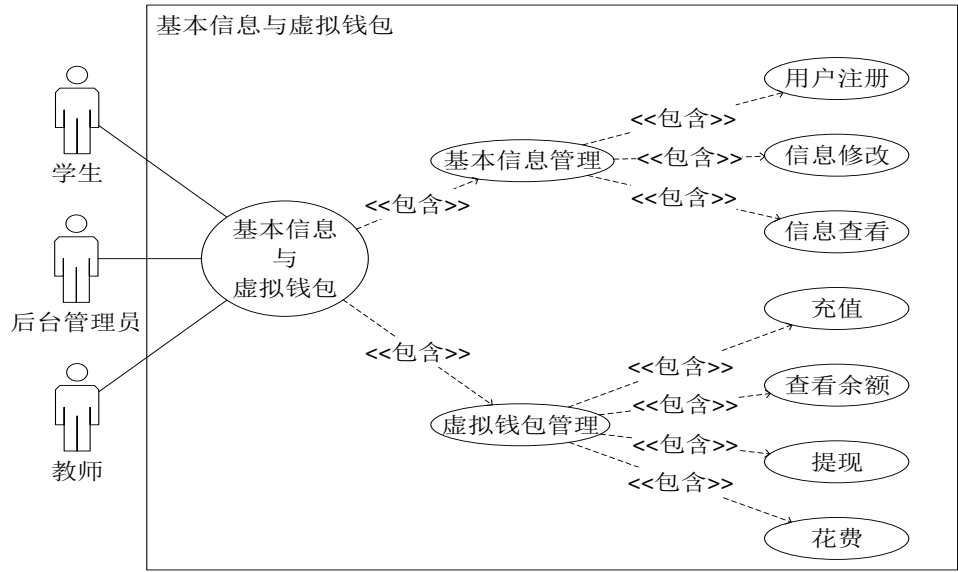


图 3-2 基本信息与虚拟钱包用例图

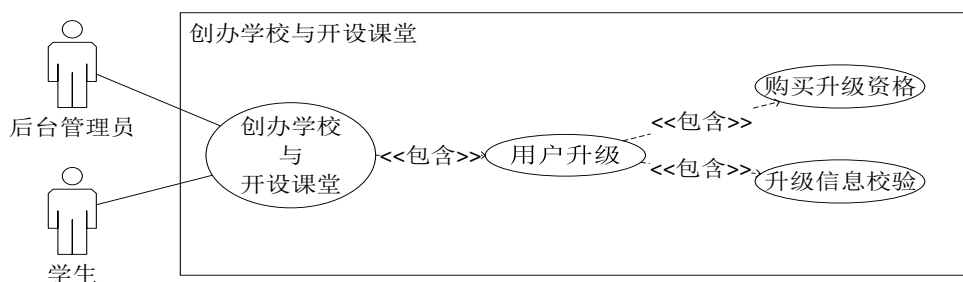
账户管理面向学生用户和教师用户提供了虚拟货币的基本管理功能，用户可以在查看自己的基本信息时浏览自己的账户余额，也可以充值、提现，学生用户在加入课堂学习之前以及用户升级过程中都需要花费虚拟货币，教师用户在创办学校和开设课堂之前也需要向系统支付一定的费用。账户管理模块涵盖了围棋直播教学系统内部基本的虚拟货币功能。在此以用户充值为例，详细描述用户对虚拟钱包充值过程中，应当如何操作。如表 3-1 所示，即为用户充值的用例描述。

表 3-1 用户虚拟钱包充值用例描述

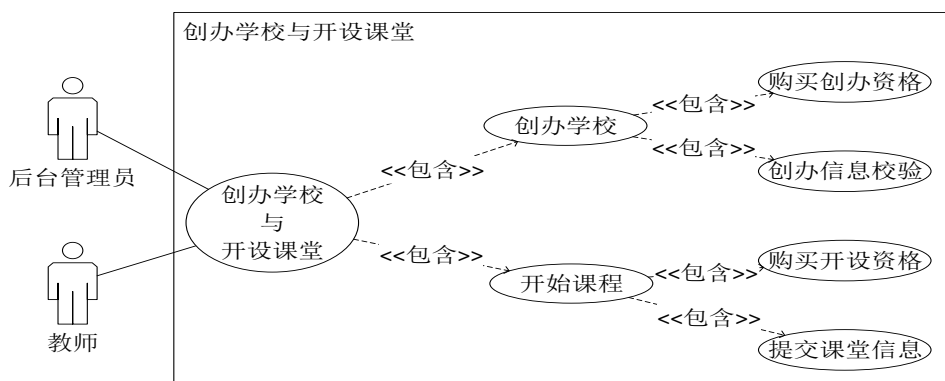
用例名称	用户虚拟钱包充值
简要说明	虚拟钱包充值，余额增加
执行者	学生或者教师用户
前置条件	用户已经登陆系统
后置条件	客户更新后的信息保存到数据库中，充值后余额
基本流程	1.进入用户信息界面； 2.查看账户余额，点击充值； 3.输入充值数量，点击提交，保存修改内容。 4.系统显示充值后余额。
扩展流程	充值中出现非法字符，系统提示充值失败。

3) 创办学校与开设课堂用例分析

创办学校与开设课堂作为围棋直播教学系统最为核心的需求之一，面向学生用户、教师用户。学生用户参与升级，教师用户参与学校创办以及课堂开设。后台管理员负责审核用户升级的资料，维护学校和课程数据。如图 3-3 所示为创办学校与开设课堂用例图。



a) 学生用户升级用例图



b) 教师用户创办学校与开设课堂用例图

图 3-3 创办学校与开设课堂用例图

由图 3-3 中 a) 图可知, 用户升级主要针对学生用户, 用户升级后成为教师用户, 拥有后续的创办学校和开设课堂的权限。用户升级需要向系统购买相应资格, 用户获取升级资格后, 提出转换身份的申請, 提交可以证明围棋办学资历的相关证明, 系统审核通过后学生用户升级成功, 之后登陆系统应当自动识别为教师用户。由图 3-3 中 b) 图可知创办学校针对教师用户, 创办学校同样需要向系统购买相应资格, 教师用户获取资格后, 可以提交创办的学校资料, 例如学校所属的段位、学校 Logo、教学目标等等, 系统做出判断, 无重复学校则创办成功, 之后教师登陆系统自动加载所创建的学校。开设课堂只针对教师用户, 且教师用户应当已有创办的学校, 开设课堂同样需要向系统购买相应资格, 教师用户获取资格后, 可以在对应学校内提交开设课堂的资料, 例如课程所属的段位、课堂 Logo、开课时间等等, 系统做出判断, 同一学校内无重复课堂则创办成功, 之后进入学校主页应自动加载已经开设的课堂。新课程开设之后系统会返回给教师用户一个推流地址, 教师用户在指定时间段内使用 `ffmpeg` 等软件将直播流推送到指定地址即可完成课堂直播。具体的课程开课时间以及直播的推流地址, 教师用户也可在课程管理模块中查看。

用户升级模块中, 学生提交相关信息之后, 需要等到后台管理员审核, 审核通过后直接成为教师用户, 否则仍为学生用户, 但仍拥有升级资格, 无需重复购买。提交证明材料是为了保障围棋教学质量, 防止围棋教学水平参差不齐。创建学校模块中, 教师用户每次只可购买一次建校资格, 创建学校成功则建校资格自动消失。若系统判定为建校失败, 则建校资格任然保留, 需重新提出申请。开设课堂模块中, 教师用户每次只

可购买一次开课资格，开设课堂成功则开课资格自动消失。若系统判定为开课失败，则资格任然保留，需重新提出申请。教师用户在开设课堂环节中，须选择课堂所属的学校，即没有创办的学校无法直接开设课堂。每所学校内可允许开设多门课程，并无限制。同样，教师用户也可开设多所学校，系统不做限制。学校为教师用户上传的课件等资料提供了统一管理统一分享的作用，课堂为教师用户开展不同教学内容提供了一定的隔离性，不同课堂内容对应可设不同的收费情况，也更加贴合围棋教学的实际情况。在此，以教师用户开设围棋课堂为例，描述教师与系统的交互过程。如表 3-2 为教师用户开设新课堂用例描述。

表 3-2 教师用户开设课堂用例描述

用例名称	教师用户开设课堂
简要说明	教师用户开设新课堂
执行者	教师用户
前置条件	用户已经登录系统，创办了学校
后置条件	系统在学校管理页面显示新开设课程
基本流程	<div>1. 进入用户信息界面；</div> <div>2. 选择已开设学校，进入学校管理页面；</div> <div>3. 点击开设新课程；</div> <div>4. 若无开课资格，系统会自动进入支付页面，花费指定数量的虚拟货币购买资格，余额不足系统提示购买失败；</div> <div>5. 在开设新课模态框中输入开设新课所需资料；</div> <div>6. 点击提交，等待审核；</div> <div>7. 系统检查有无重复课程课程重复，开课失败；</div> <div>8. 对应学校管理页面显示已开课程。</div>
扩展流程	用户余额不足或者开课资料不足，系统提示失败；

4) 围棋资源与参与课堂用例分析

在线课堂学习是围棋直播教学系统最为核心的需求之一，资源的获取以及课堂教学课堂，都需要学生用户和教师用户的参与。后台管理员负责维护功能模块中的数据。如图 3-4 所示为围棋资源与参与课堂用例图。系统为用户提供了观看视频自学和加入直播课堂进行教学学习两种围棋学习方式。

具体在自学模块中，系统提供了丰富的免费围棋教学视频，并将这些视频划分到不同的段位，用户可以根据自己的需要选择进入具体的段位观看相应水准的围棋视频。自学模块的视频属于围棋资源的一部分，由系统提供，所有用户均可观看。

加入课堂学习是围棋资源与参与课堂最为主要的功能。课堂学习针对学生用户和教师用户均提供了段位选择、直播学习、课堂讨论、课件与作业的上传下载等功能。用户登陆系统后，可以选择合适段位的学校与课堂，通过购买学习资格后进入课堂。教师用户默认对自己开设的课堂拥有权限，无需购买。进入课堂之后，用户可以观看课堂直

播，课堂直播需要开设课堂的教师用户通过推流软件进行推流直播，推流地址在校园管理与课程管理模块中指定。

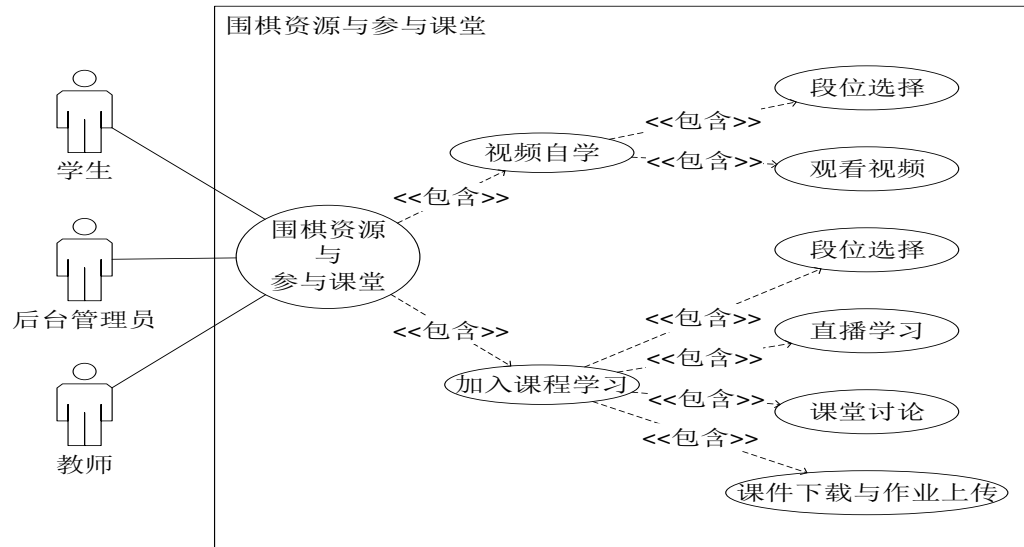


图 3-4 围棋资源与参与课堂用例图

课堂教学直播过程中，教师可以通过教师用户身份与学生用户在线交流，进行课堂提问，学生用户之间也可以互相讨论。围棋资源服务功能中，为了学生用户和教师用户最大限度的利用资源，还提供了文件上传下载功能，教师可以上传教学课件供学生用户提前预习和课堂参考，学生用户也可以提交作业，教师批改后回复学生。围棋资源还包括学校内文件共享，学生可下载参与课堂所属学校的校园文件。在此，以学生用户观看教学直播为例，描述学生用户、教师用户与系统的互动性。如表 3-3 为观看直播课堂的用例描述。

表 3-3 观看直播课堂用例描述

用例名称	观看直播课堂
简要说明	上课过程中，学生通过浏览器页面观看教师直播
执行者	学生和教师用户
前置条件	用户已经登录系统
后置条件	学生可实时观看教学视频
基本流程	1. 选择参与课程段位，系统提供学校推荐； 2. 在学校主页页面选择课程，点击进入； 3. 若用户无观看权限，系统提示支付一定量的虚拟货币，余额不足则购买失败，无参与课堂权限； 4. 进入课堂页面，自动获取拉流地址。教师用户通过 ffmpeg 等软件将视频流推送到 RTMP 地址； 5. 在 JW Player 播放直播画面。
扩展流程	学生用户若没有参与课堂权限，需要先行购买课堂参与资格，否则无法进入课堂观看直播



### 5) 校园管理与课程管理用例分析

课程管理作为围棋直播教学系统最为核心的需求之一，主要面向学生用户、教师用户，学生用户管理自己的课程表，教师用户管理开设的学校和课程。后台管理员负责维护校园和课程数据，以及校园内文件，课堂内的课件与作业数据。如图 3-5 所示为校园管理与课程管理用例图。

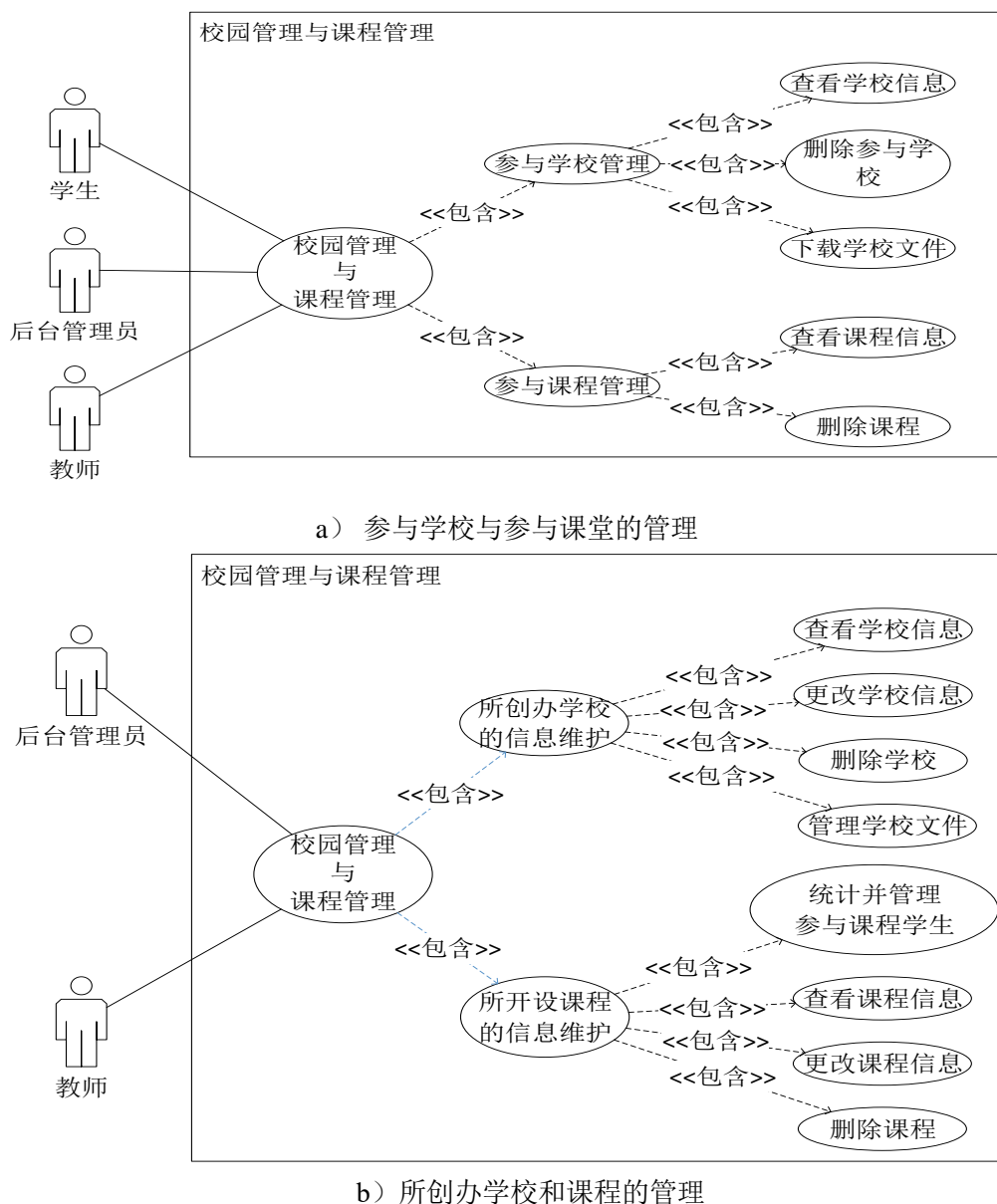


图 3-5 校园管理与课程管理用例图

由图 3-5 中 a)图可知，学生用户只可查看参与的学校与课程信息，并对自己参与的课程表有删除权限。图 3-5 中 a)图可知，校园管理与课程管理功能需求中教师用户主要参与校园信息维护和课程信息维护两大部分。由于学校和课堂由教师用户付费建立，教师用户对其拥有全部的管理权限，具体包括学校信息维护中的查看学校信息、更改学校信息、删除学校、管理学校文件，课堂信息维护中的查看课程信息、更改课程信息、删除课程。

删除课程，统计并管理参与课程学生。

教师用户删除学校时，校内课程信息自动全部删除。教师用户删除课程时，购买对应课程的用户课程表中，此课程自动删除。并且若学生用户只参加了某学校内的一门课程，此课程删除后，该学生用户也无权限查看该学校主页内的共享文件。此外，课程管理功能中，教师用户对参与课程的学生有管理权限，若课堂中有人捣乱或散步虚假消息，可以剔除该破坏分子。此功能也为学校统计参与人数提供了方便。在此，以教师用户删除开设课程为例，描述教师用户与系统的互动性。如图 3-3 为教师用户删除课程的用户例描述。

表 3-4 教师用户删除课程用例描述

用例名称	教师用户课程
简要说明	教师用户进入课程对应的学校主页，删除相应课程按钮
执行者	教师用户
前置条件	用户已经登陆系统，对应课程已开通
后置条件	学校主页中，不在显示该课程信息；所有学生的参与课程中，该课程消失；只参与此课程的学生无法下载对应学校的文件
基本流程	<ol style="list-style-type: none"> <li>1. 进入教师用户个人中心；</li> <li>2. 选择该课程对应学校，进入学校主页；</li> <li>3. 在学校所开设课程中选择该课程；</li> <li>4. 点击删除该课程按钮；</li> <li>5. 系统在数据库中完成此课程的删除；</li> <li>6. 学校主页不再显示该课程。</li> </ol>
扩展流程	课程删除后，无法复原，所以该课程相关信息被清除

### 3.2.2 围棋直播教学系统的结构模型

这一小节主要描述基于 RTMP 流媒体技术的围棋直播教学系统的结构模型，为了更清楚的表达系统内部实体的内部属性以及实体之间的关系，本节采用 UML 用的类图来阐述系统的静态结构。如图 3-6，即为线上教学系统的结构模型，通过对系统建模时提取的关键性的业务实体及其顶层类的抽象，可以更清楚的展示系统的静态结构。

1) `models.Model` 是 Django 框架提供的实现了 ORM 机制的数据基类，是所有基于数据库的 python 对象的父类。其中提供了 `save()`, `filter()`, `delete()` 等方法，实现了数据的存储、查询、更新以及删除。

2) `views.generic.View` 是 Django 框架中用于处理所有请求的视图类，用户的所有请求都会被 `dispatch` 方法接受，然后根据不同的 http 请求进行分发，`get` 请求会被转发到 `View` 子类对象的 `get()` 函数，所有 `post` 方法的请求会被转发到 `View` 子类对象的 `post()` 函数。

3) `User` 类继承自 `Model` 类，是系统两类主要用户的基类，其中包含了用户的昵称，登陆密码，邮箱等共同的基本信息，`User` 类的 `umoney` 属性对应两类主要用户的虚

拟钱包，用户基本的充值、花费与提现功能基于此属性实现。

4) Ustudent 继承自 User 类，除基本信息之外，由于学生用户转为教师用户需要购买资格，因此学生用户的 `can_upgrade` 属性决定了学生用户是否拥有升级权限，默认没有。学生用户需花费一定量的虚拟货币向系统购买资格。

5) Uteacher 继承自 User 类，除基本信息之外，教师用户的 `can_createschool` 属性决定了教师用户是否拥有创办学校的权限，默认没有。教师用户需花费一定量的虚拟货币向系统购买资格，教师用户建立学校后，此属性应当重置为否。教师用户在升级之时还需要补充相关的围棋段位证明和教学资历证明，对应 `uTeach_credential` 和 `uGo_credential` 属性。

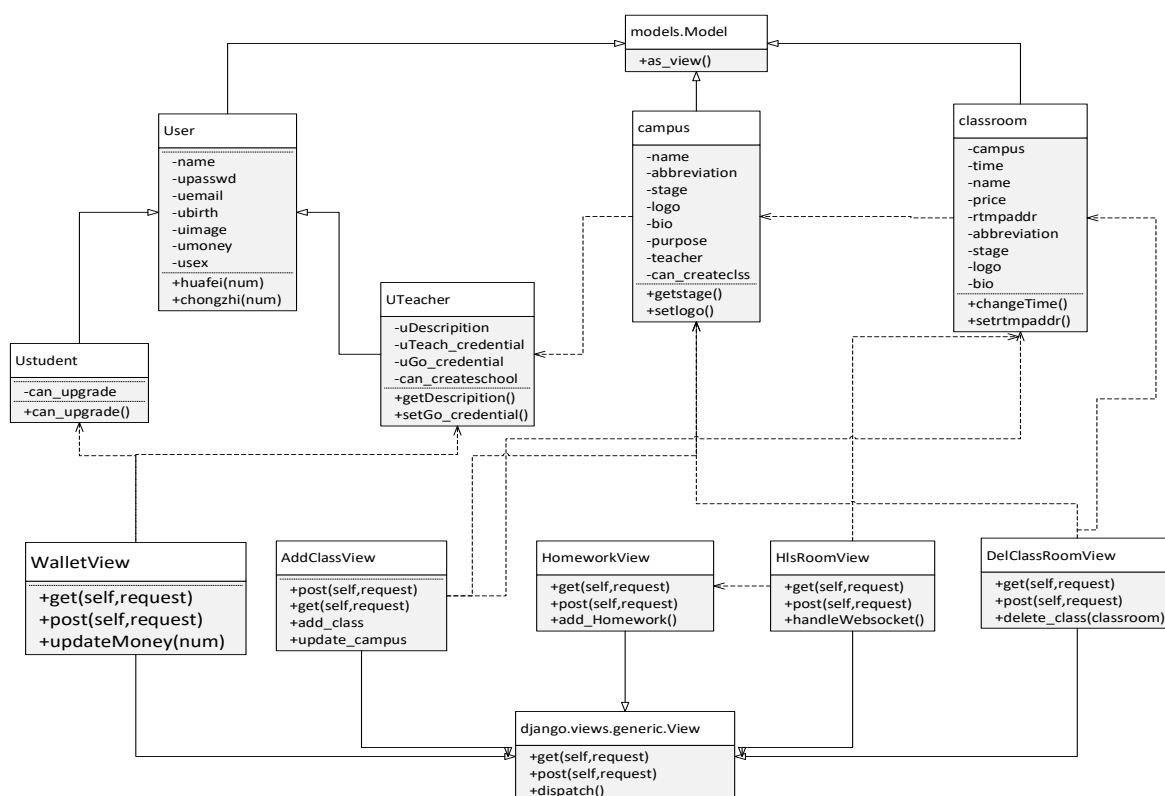


图 3-6 系统总体类图

6) Campus 直接继承自 Model 类，其中属性均为校园基本信息，例如学校简称，办学宗旨等等。Campus 类的 `can_createclass` 属性决定了此校园下教师用户是否拥有设立新课堂的权限，默认没有。教师用户需花费一定量的虚拟货币向系统购买资格，建立课堂后此属性应当重置为否。

7) Classroom 直接继承自 Model 类，其中属性均为课堂基本信息，例如课堂简称，所属段位等等。Classroom 类的 `price` 属性指定了其他用户参与此课堂的价格，需付费参与学习，其 `rtmpaddr` 属性由系统指定，不同课堂依据学校学校简称与课程简称的组合划分不同的推流地址，实现了直播流的分离，`time` 属性由教师用户填写，代表了开课时间，在此时间段内教师直播推流，学生拉流，即实现了直播教学。

8) WalletView 继承自 View 类，接受到用户充值或提现的需求后，由 `post` 方法取

出虚拟钱包的更新信息，调用 `updateMoney` 处理虚拟货币的增减。余额不足或充值过多，均应当提示。

9) `AddClassView` 继承自 `View` 类，教师用户发送创建课堂的请求后，由 `post` 方法调用 `add_class` 函数，传递新建课堂的属性信息，存入数据库中完成信息更新。课堂创建成功后，应当重置对应 `Campus` 的 `can_createclass` 属性，对应 `updateCampus` 方法。

10) `HomeworkView` 继承自 `View` 类，负责处理课堂内教师课件以及学生作业的上传与下载。

11) `DelClassRoomView` 继承自 `View` 类，教师用户发送删除课堂的请求后，由 `post` 方法接受请求，过滤出课堂的信息，通过 `delete_class` 方法实现系统内该课堂信息的删除，以及参与此课程的用户课程表内该课程的删除。

### 3.2.3 围棋直播教学系统的行为模型

通过对围棋直播教学系统功能模型的分析，可知系统总体分为基本信息与虚拟钱包、围棋资源与参与课堂、创办学校与开设课堂和校园管理与课程管理四个功能需求模块。下面分别具体描述这些需求的行为模型，并通过活动图的形式展现用户与系统间的交互行为。

#### 1) 基本信息与虚拟钱包行为模型

基本信息与虚拟钱包为学生用户和教师用户提供了在使用系统的过程中对用户身份基本信息和虚拟钱包的维护，账户余额可以在系统中进行消费。在此以用户账户充值为例描述其行为模型，图 3-7 为用户账户充值活动图。

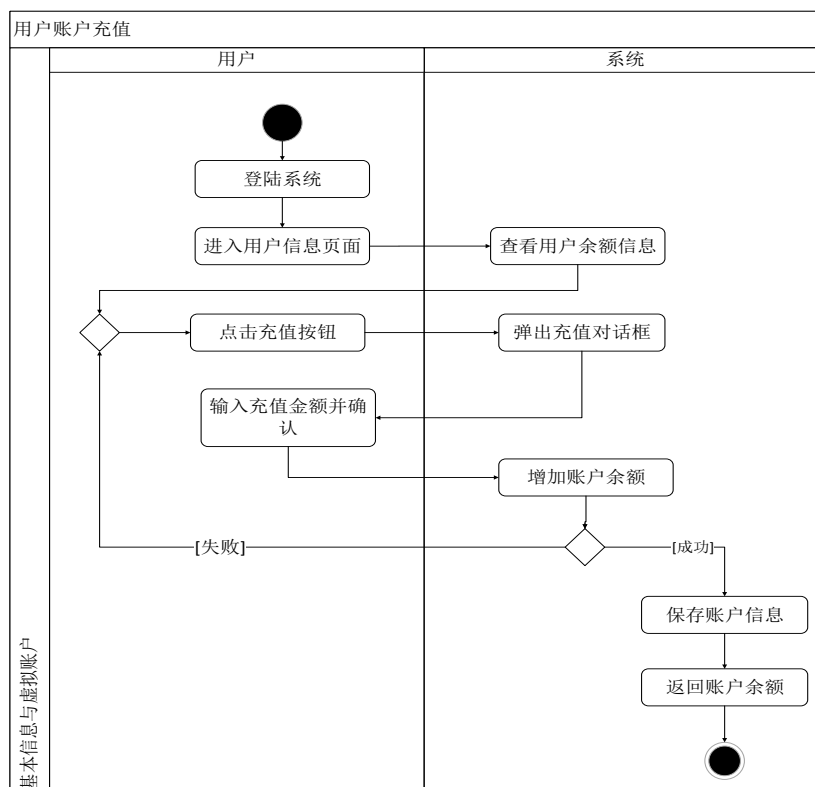


图 3-7 用户账户充值活动图

## 2) 创办学校与开设课堂行为模型

创办学校与开设课堂是指学生用户通过购买权限和资历审查升级为教师用户、教师用户通过购买建校资格创建学校以及教师用户通过购买开课资格开设新课堂的功能模块。在此以教师用户开设新课堂为例描述其行为模型，如图 3-8 所示即为开设新课堂的活动图。

由图可知，教师用户登陆系统后，应首先进入个人主页，选择新开设课堂所对应的学校，进入该学校主页，学校主页内会显示该校园的基本信息以及校园内的所有课程。点击课程管理最下面的开设新课堂按钮，系统会审查用户有无开课资格。若没有，则会弹出支付页面，花费指定数量的虚拟货币即可拥有一次开课资格，课程创建成功，该资格失效，课程创建失败，此资格保留。支付成功后，用户重新点击开设新课程，会弹出创建课程的模态框，在其中输入课程信息，点击提交。系统会审查此课程的信息是否与已有课程重复，审查通过，系统会提示开课成功并向用户提示课程的推流地址，审查失败，系统也会由“开课失败”的警示。课程创建成功后，购买此课程的用户在课程指定的开课时间，进入课程即可观看课堂直播，参与课堂互动。

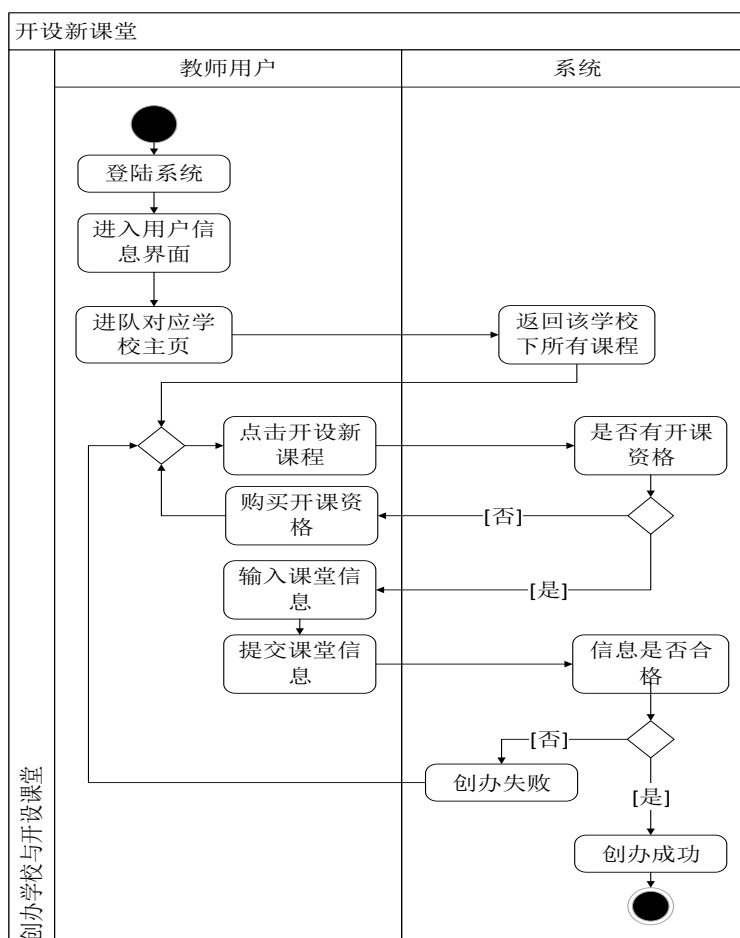


图 3-8 开设新课堂活动图

由图 3-8 可知，在用户账户充值活动图中，学生用户或教师用户均应当支持充值操

作。整个操作活动的具体流程如下：用户登陆到系统中，进入用户信息界面，用户可以查看到自己的账户余额。点击余额旁边充值按钮，系统弹出充值对话框，输入充值金额，点击确认，系统作出处理，充值成功则系统数据库更新，显示新的账户余额，充值失败弹出“充值失败”对话框，账户余额不变，需重新充值。

3) 围棋资源与参与课堂行为模型

围棋资源与参与课堂是指学生或教师用户通过本系统进行学习，包括观看视频的自学形式和参与课堂的教学式学习形式。其中，参与课堂是围棋直播教学系统最为核心的需求。参与直播课堂，用户需要采用观看直播的方式。在此以学生用户参与直播课堂为例描述围棋资源与参与课堂的行为模型，如图 3-9 所示即为参与课堂的活动图。

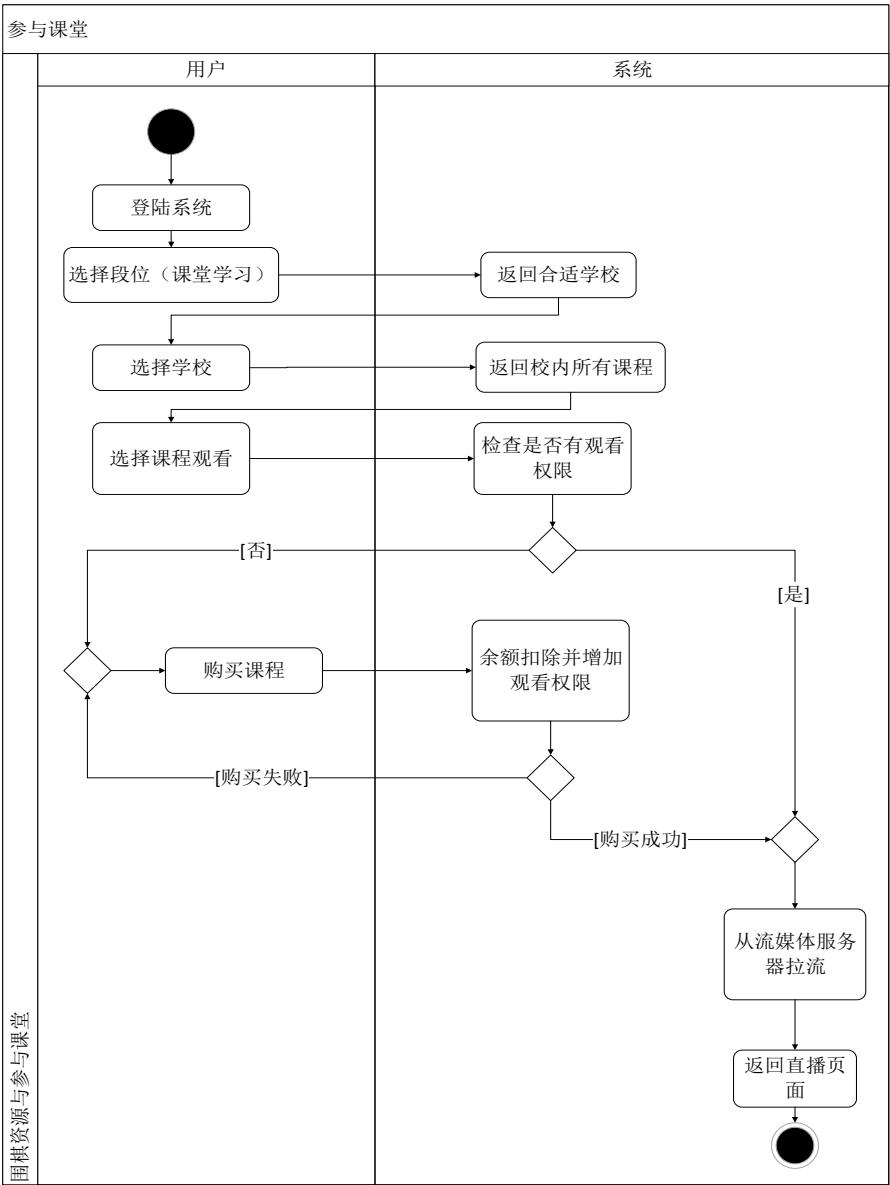


图 3-9 参与课堂活动图

参与课堂活动图中，操作对象包括学生用户和教师用户，主要是针对参与围棋学

习的学生。整个操作活动的具体流程如下：用户登陆到系统中，在相关页面选择合适的段位，系统会返回符合条件的学校，选择感兴趣的学校进入该学校主页。系统会展示该学校内开通的所有课程及相关的课程简介、开课时间、课程标价等，用户选择合适课程，点击参与课程。如果用户没有进入该课程的权限，系统会弹出购买页面，用户花费指定的虚拟货币后拥有参与课堂的权限，进入课堂学习环节。课堂的直播学习中，直播页面会依据学校与课堂信息，自动从流媒体服务器拉流，直播流由教师用户通过 `ffmpeg` 等软件采集摄像头等信息推流到流媒体服务器，至此用户可以观看到围棋的直播教学环节。

#### 4) 校园管理与课程管理行为模型

校园管理与课程管理为学生用户和教师用户提供参与课程的管理服务，包括参与课程的信息查看，用户课程表内的课程删除。并且，系统单独为教师用户提供所开学校和课程的更加高级的管理服务，包括学校和课程的增删改查。在此以教师用户删除课程为例描述其行为模型，图 3-10 为开删除课程活动图。

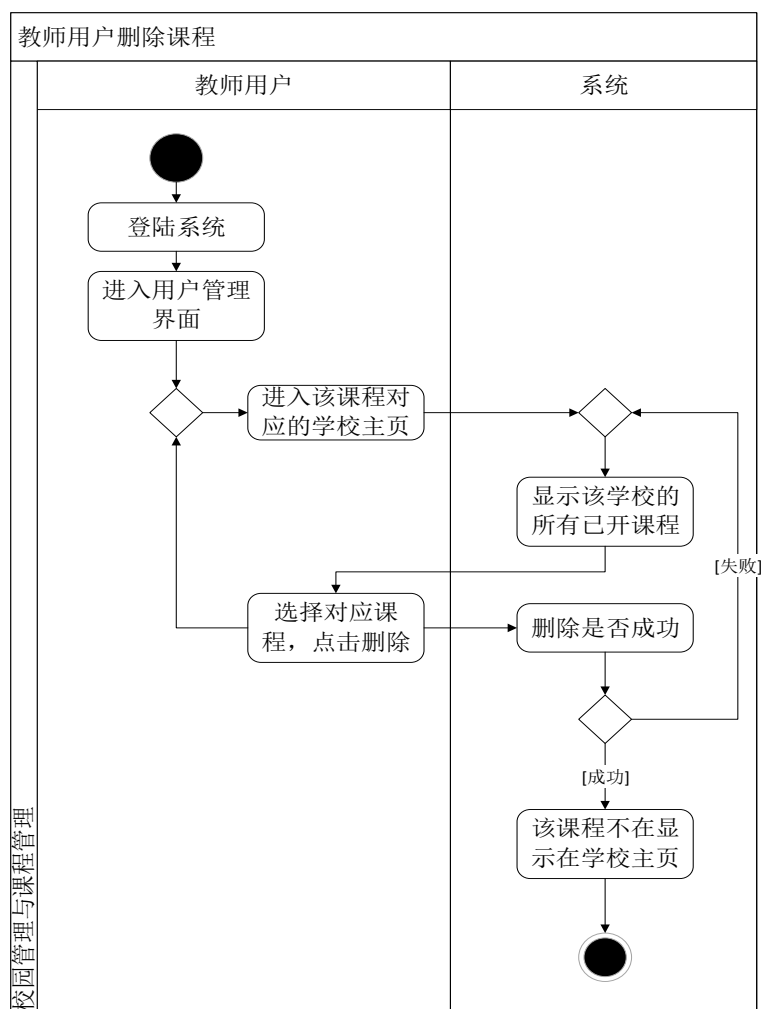


图 3-10 教师用户删除课程活动图

由图可知，教师用户登陆系统后，应首先进入个人主页，选择所删除课堂对应的学

校,进入该学校主页,学校主页内会显示该校园的基本信息以及校园内的所有课程。查找所删除的课堂,点击删除,系统会删除此课程相关的所有信息,并在参与此课程的用户课程表中移除此课程记录。删除失败,系统会有相应提示;删除成功,页面自动刷新,不再显示该课程信息。

### 3.3 围棋直播教学系统的非功能需求

结合围棋直播教学系统的功能特点和实际需求,一些非功能性的需求也会影响到的使用,尤其是用户对于直播课堂的时延性有着较为严格的需求。因此本节将对系统的非功能需求从兼容性需求、直播时延需求、承受压力需求三个方面进行分析。

1) 兼容性需求。由于用户可在手机、平板、PC 等不同平台参与课堂,所以应当测试用户在不同的机型、不同的系统、不同的分辨率甚至不同网络环境下是否可以正常进入课堂直播间观看直播。

2) 直播时延需求。用户在观看直播时,过长的时延将影响到教师用户与学生用户的交流。考虑到直播课堂中,基本以教师用户长时间的讲解为主,直播系统的延时在可接受范围内。

3) 承受压力需求。直播课堂的压力测试即同一课堂的并发访问数量需求。同一课程内,应允许多名学生用户同时观看直播而不影响直播效果,即在直播过程中,应保证所有参与该课程的同学均可以较为流畅的获取直播流。

### 3.4 本章小结

本章研究了围棋直播教学系统的总体需求,分析了完成围棋直播教学系统所需要的系统角色、围棋直播教学系统的大概模块划分、,以及系统角色与教学系统之间的交互行为。通过需求描述和用例图可知围棋直播教学系统需要学生用户、教师用户、系统后台管理员共同参与完成基本信息与虚拟钱包、在线课堂学习、创办学校与开设课堂、课程管理四个服务模块,并通过用例描述分析了其中的部分典型用例;并通过系统的结构模型和行为模型来更清楚的展示系统的整体架构和业务流程;最后完成了系统非功能需求的介绍。通过系统分析,为后面的详细设计提供了依据和支持,使系统的设计部分更加清楚简洁。



## 4 基于 RTMP 流媒体技术的围棋直播教学系统的设计

通过上一章中对系统的需求描述以及对系统相关功能模型、结构模型和行为模型的分析，系统的需求已经十分清楚。本章在此基础上完成系统的整体架构和功能模块划分，并对各功能模块具体的设计展开详细描述。最后对系统的主要数据库完成详细设计，包括通过 E-R 图表达的概念数据模型以及说明表内字段的物理数据模型。

### 4.1 围棋直播教学系统的概要设计

#### 4.1.1 围棋直播教学系统的软件体系结构

围棋直播教学系统为 B/S 模式，学生用户只需安装浏览器，通过 Internet 即可访问本系统；教师用户开设课程的推流环节需要将视频流推送到课程指定的 RTMP 地址，例如可以借助 ffmpeg 等将摄像头采集的视频流推流到指定地址，除此之外不需要其他的额外软件。由之前的系统需求分析可知，系统设计共分为三层：数据支持层，业务逻辑层和用户界面层。系统的软件体系结构如图 4-1 所示。

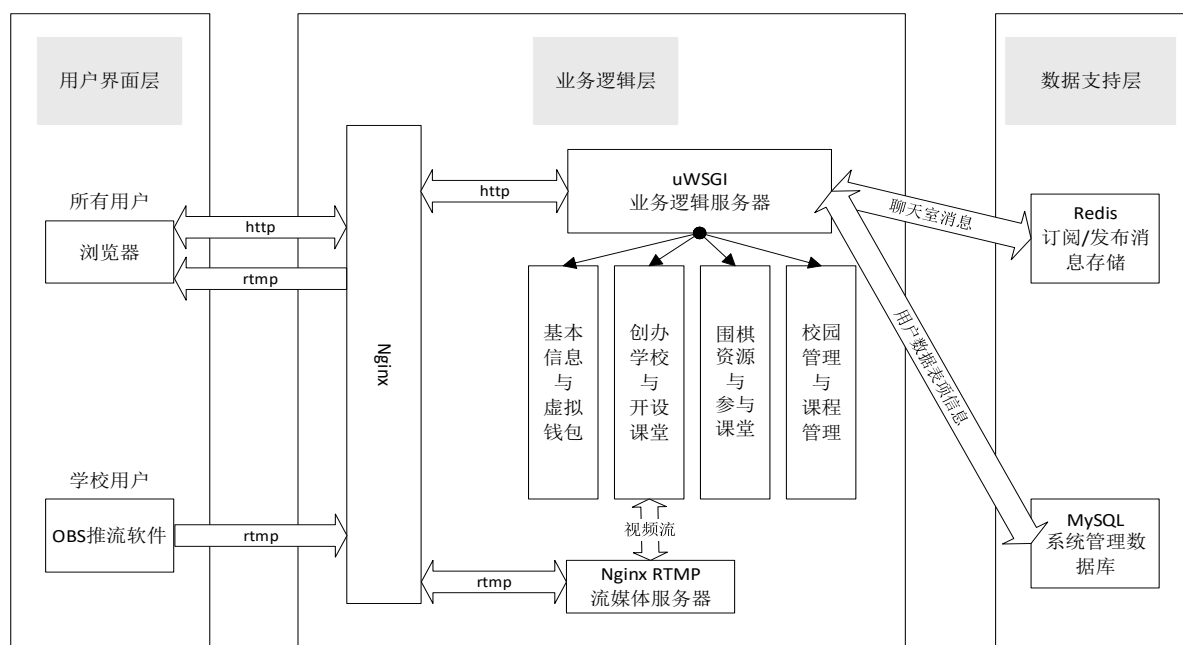


图 4-1 系统软件体系结构设计

#### 1) 数据支持层

数据支持层主要用于完成系统对数据库的基本操作，以及对课堂互动时聊天信息的发布和订阅的支持。对于数据库的基本操作包括对用户信息、学校信息以及课程信息的增删改查，聊天信息的发布和订阅则通过订阅 Redis 的相关频道完成。数据支持层功能的完善为业务逻辑层提供了健全的服务，业务逻辑层的数据读取和数据写入依托于数据支持层。

#### 2) 业务逻辑层

业务逻辑层定义了系统的业务流程和服务逻辑，既可处理基本的用户请求，同时完成了流媒体服务器的功能。用户的请求主要包括基本信息与虚拟钱包，围棋资源与参与课堂，创办学校与开设课堂，校园管理与课程管理 4 大功能模块的业务导向。

流媒体服务器的功能是接受教师的直播流，学生用户在观看时可以直接通过浏览器拉流观看。学生用户无需关心如何拉流，系统会通过 JavaScript 程序判断用户所选课程，自动拉取指定视频流。

### 3) 用户界面层

通过浏览器与用户交流，完成信息输入以及数据的接收与展示。教师用户还需借助推流软件，将采集的视频流推送到指定地址。

## 4.1.2 围棋直播教学系统的功能模块结构

在系统分析的基础上可知，系统从整体功能上分为基本信息与虚拟钱包，围棋资源与参与课堂，创办学校与开设课堂，校园管理与课程管理四大模块，其功能结构如图 4-2 所示。

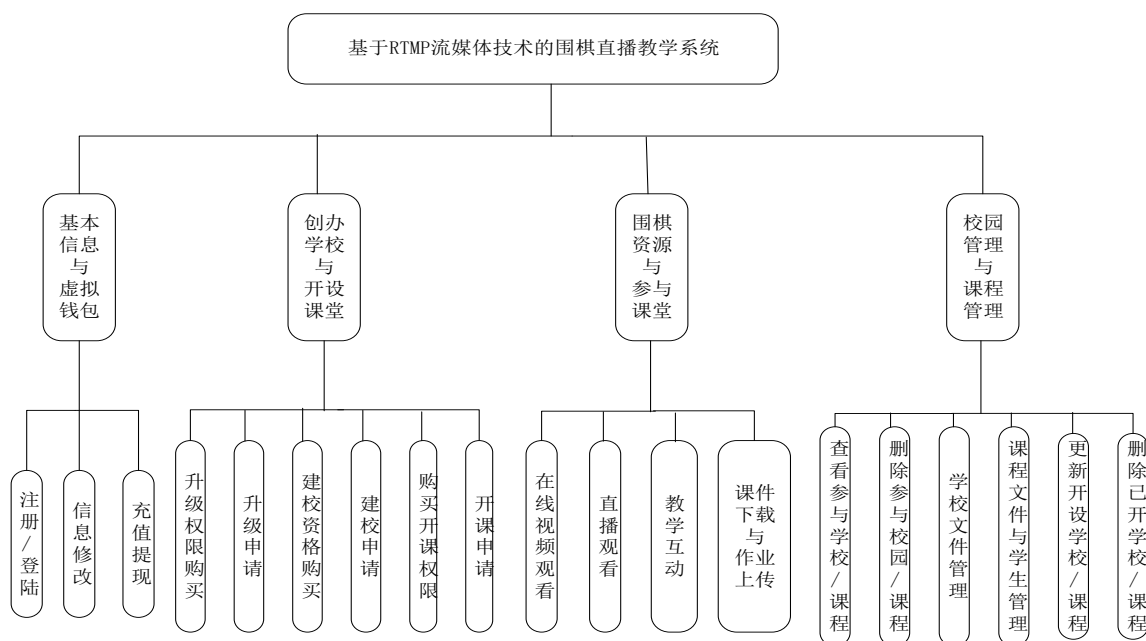


图 4-2 系统功能模块结构

从图 4-2 可以看出，各模块的逻辑服务具体如下：

基本信息与虚拟钱包：用户的注册、登陆、信息更新以及用户钱包的充值和提现；

用户升级与创建学校：学生用户选择购买升级权限并升级为教师，教师用户购买建校资格并创办学校，教师用户购买开课资格并开设新课堂；

围棋资源与参与课堂：用户选择段位观看视频，或根据段位进入直播课堂，课堂内的讨论交流和课件下载，作业上传；

校园管理与课程管理：学生用户和教师用户对于参与课程的查看与删除，教师用户对所开设学校与课堂的信息更新以及删除，同时包括了校园内的文件管理和课堂中的

学生管理与统计。

## 4.2 围棋直播教学系统的详细设计

在上一小节说明了系统的软件架构和功能模块划分，以及各功能的作用和提供的服务。在此基础之上，本节将对各功能模块进行详细设计。通过分析个模块内部的静态结构和模块内部类之间的信息交互，可以更详细的展现模块内部的设计架构。

### 4.2.1 基本信息与虚拟钱包

#### 1) 静态结构详细设计

基本信息与虚拟钱包是系统最基础的部分，这两个小的功能模块分别维护了用户最为基础的信息内容和虚拟钱包的内容。用户身份主要由学生和教师组成，分别维护着不同的身份信息；虚拟钱包是两类用户都必须的，是用户在其他模块接受服务所必需的。该模块的类图如图 4-3 所示。

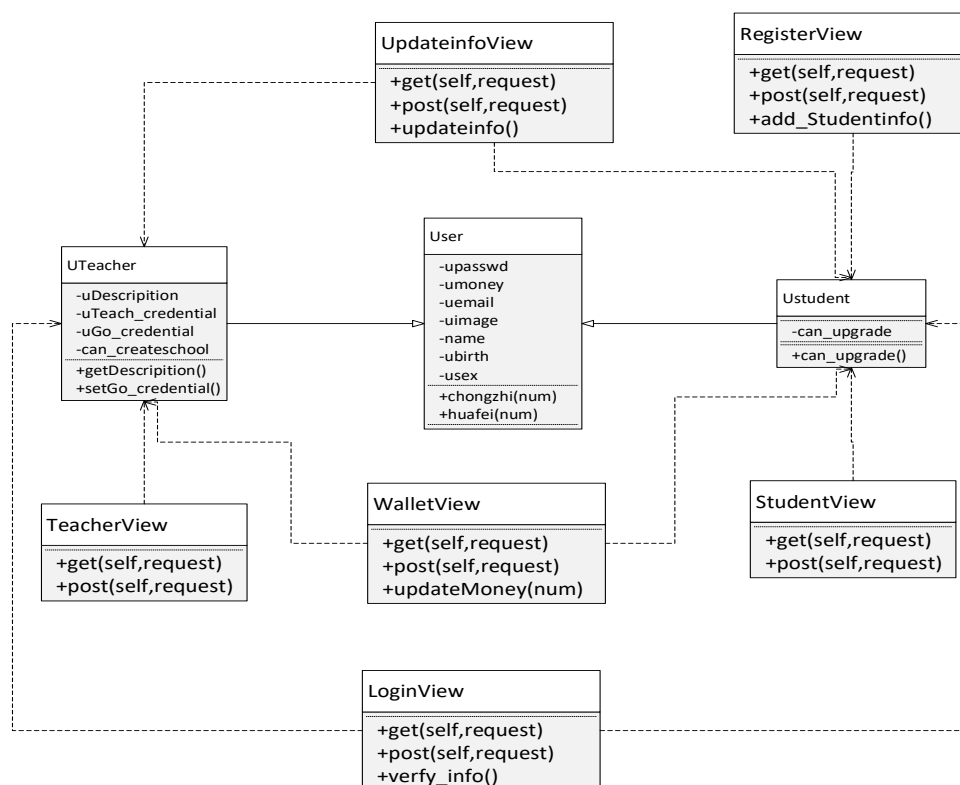


图 4-3 基本信息与虚拟钱包类图

由图可知，本系统主要面向学生和教师用户，这两类用户有着很多类型的基本信息，例如昵称、性别，邮箱等，系统通过抽象成 **User** 类来维持基本的信息。由于学生用户和教师用户在其他模块所需的权限不同，因此分别从 **User** 类继承之后维护了不同的权限，**Ustudent** 类对应学生用户，**Uteacher** 类对应教师用户，两类用户的 `umoney` 属性维护了他们的虚拟钱包的信息。系统通过 **RegisterView** 类来接受用户的注册，在

`add_student()`方法中用注册信息实例化 `Ustudent` 类实现新用户的添加,默认注册用户都是学生,教师用户添加通过用户升级模块完成;用户通过 `LoginView` 类提交登陆信息, `verify_info()`函数查询相关用户密码,通过验证后进入系统享受提供的服务; `StudentView` 和 `TeacherView` 可以通过查询登陆用户的具体信息,反馈到浏览器页面,实现用户对于自己基本信息的查看; `UpdateinfoView` 在用户点击提交更新信息后,通过 `post()`方法调用 `updateinfo()`函数,实现数据库内用户信息的持久更新,完成了用户修改基本信息的服务;账户管理的功能通过 `WalletView` 完成,系统通过调用 `updatemoney()`方法获取充值或提现信息,完成用户账户的充值与提现,其中提现时若余额不足会有提示,充值时数量过大也应当提示。

## 2) 交互详细设计

上面通过对模块内类的说明分析了类内部提供的服务以及类间的交互和调用接口,下面采用时序图的方式详细说明模块内部的信息交互。如图 4-4 是用户充值时的类间交互。

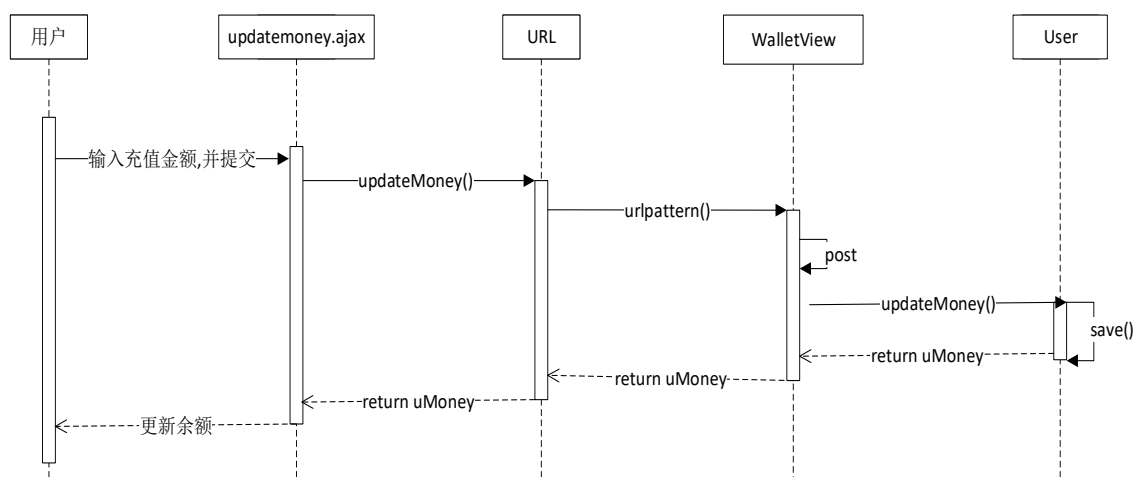


图 4-4 用户充值时序图

用户充值时,会触发用户中心页面的 `ajax` 函数,向系统提交 `post` 请求, `django` 的路由分发机制会将对应的请求送到对应的类中,此处 `WalletView` 处理充值信息。`WalletView` 在 `post` 函数内通过 `updateMoney()`处理充值/提现的逻辑,通过 `User` 类的 `save()`方法更新数据库中用户的余额,并返回新的账户数据,最终 `updatemoney.ajax` 文件内函数更新用户中心页面的余额数据,或充值失败,系统会弹出提示框。

## 4.2.2 创办学校与开设课堂

### 1) 静态结构详细设计

创办学校与开设课堂使拥有围棋教学资格的用户可以在系统内开建围棋学校和课堂,将围棋资源和围棋教学的收益扩大化。用户升级针对学生用户,购买升级资格并提交申请资料,审核通过后转为教师用户;创办学校只针对教师用户。创办学校与开设课堂也需要购买相应资格,提交校园和课堂资料。创办学校与开设课堂模块的类图如图

4-5 所示。

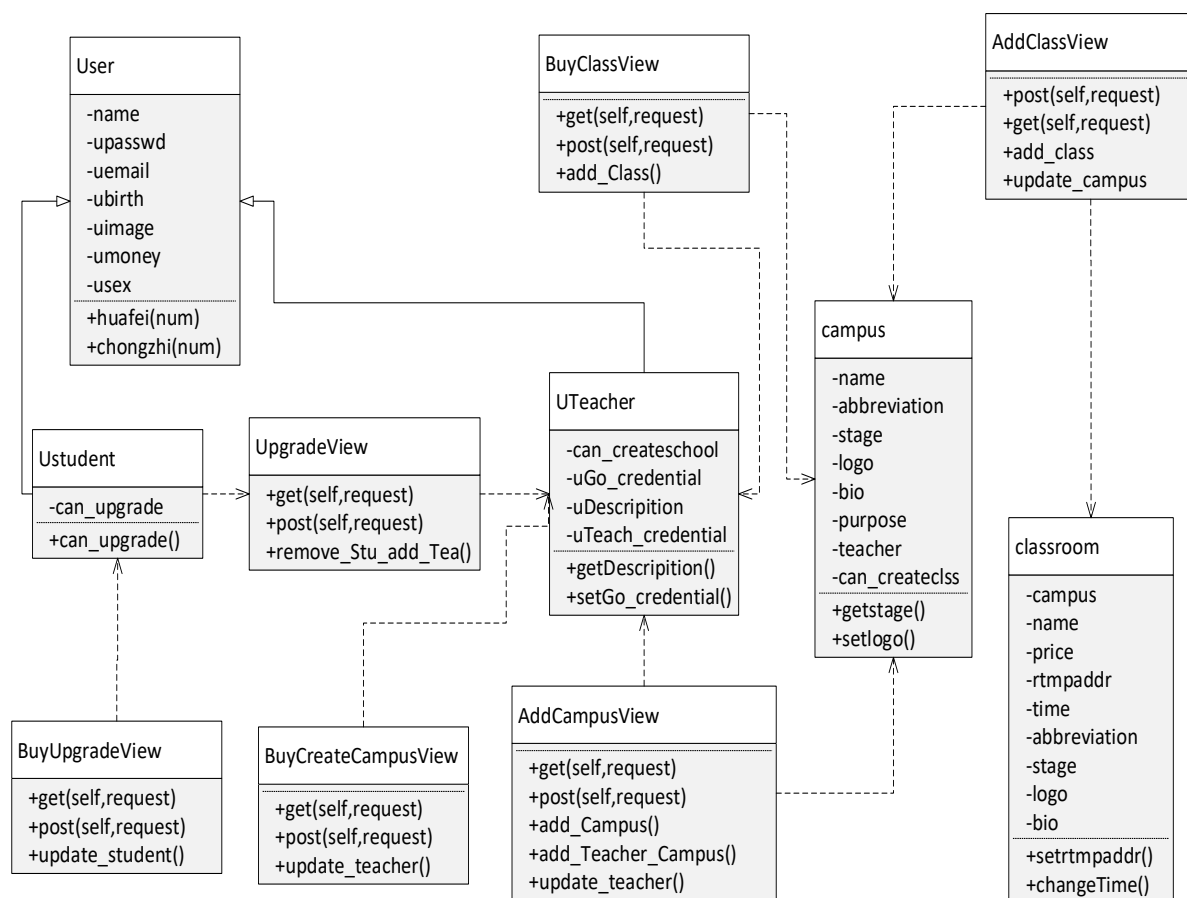


图 4-5 创办学校与开设课堂类图

由图 4-5 可知，Ustudent 和 Uteacher 均继承 User 类，其中 Ustudent 的 can\_upgrade 属性决定了学生用户是否有升级权限，Uteacher 的 can\_createschool 属性决定了教师用户是否拥有创办学校的权限，同样，Campus 类的 can\_createclass 属性决定了学校创建者是否可以在此校园内开设新课堂。BuyUpgradeView 处理学生用户升级资格逻辑，BuyCreateCampus 处理教师创建学校资格的逻辑，BuyClassView 处理校园内开设课堂资格的逻辑问题。各用户拥有对应资格后，UpgradeView, AddCampusView, AddClassView 分别对应用户升级、创办学校，开设课堂内的逻辑处理。

## 2) 类间交互详细设计

通过上面对模块内类的说明分析了创办学校和开设课堂时用户权限与服务逻辑的问题。在此以教师用户开设课堂时的序列图说明系统内部的类间交互。如图 4-6 所示。

由图 4-6 可知，教师用户在开设新课堂页面，填写课堂信息并提交后，addclass.ajax 会解析页面的课堂信息，通过 post 向系统发送请求和数据，django 的路由分发机制会将对应的请求送到对应的类中，此处 AddClassView 处理开课信息。通过类内部的 add\_class() 函数实例化 Classroom 对象，并进行数据持久化的存储。存储

成功后，校园所属的教师用户失去开课资格，重新开课需要重新购买开课权限。  
`update_campus()`方法更新课堂所属学校的 `can_createclass` 属性，返回最终状态。最终，`addclass.ajax` 会根据系统返回的状态显示最新的课堂列表。

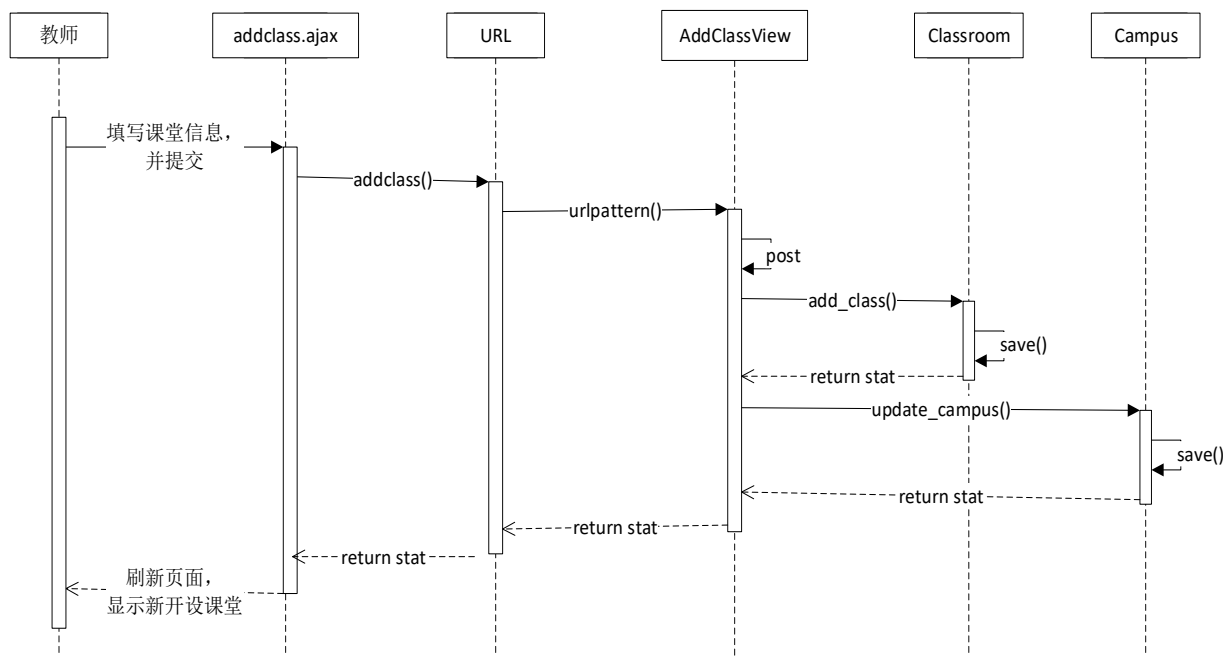


图 4-6 教师用户开设新课堂时序图

### 4.2.3 围棋资源与参与课堂

#### 1) 静态结构详细设计

围棋资源与参与课堂是围棋直播教学系统中最为重要的一部分，围棋教学是围棋直播教学系统设计的主要目的。如 4.1.2 节分析，学生或者教师用户都可通过视频观看或者参与直播课堂进行围棋学习。该模块的类图如图 4-7 所示。

围棋资源与参与课堂由 `IndexView` 返回选择界面。通过视频观看学习时，需要段位选择后，挑选感兴趣的视频进行观看，由 `video` 类通过过滤段位后返回视频地址，前端页面通过 `JW Player` 进行视频播放；参与直播课堂同样需要段位选择，之后会返回过滤后的学校，进入学校，`CampusView` 返回学校信息及校内课程。用户选择参与其中某个课程，`BuyClassView` 会判断用户是否拥有参与权限。若没有参与权限，`BuyClassView` 完成用户的购买，其中涉及到学生用户的课表添加以及学生用户向教师用户的支付。拥有参与课堂的权限之后，`HlsRoomView` 对象将负责课堂页面的直播式围棋学习，在 `get()` 方法中，会根据提交的课程信息过滤出合适的课堂信息，返回课堂的 `rtmp` 地址，前端页面可以从对应地址拉流播放。参与课堂期间，用户可以在观看直播教学的同时，`HlsRoomView` 的 `handleWebsocket` 函数将实例化 `RedisHelper` 对象，实现 `Redis` 的消息订阅与发布，同一课堂的用户将接受相同频道的信息，同时也可发送信息到该频道，实现了直播期间的用户交流，`HomeworkView` 将负责用户下载教师上传的课件，并上传自己的作业。

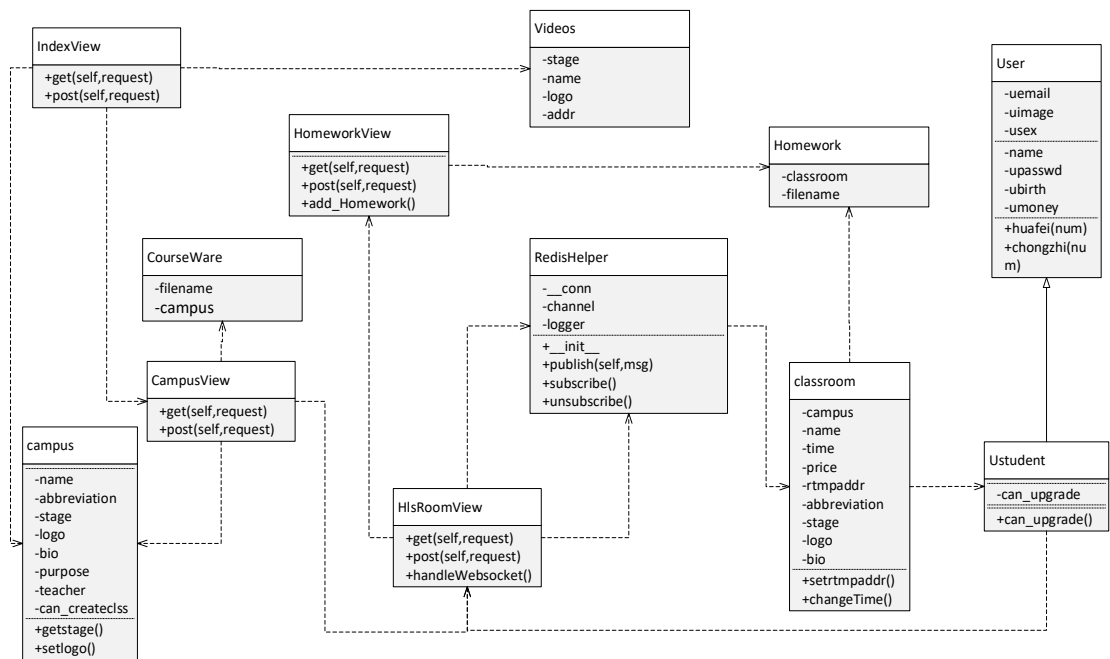


图 4-7 围棋资源与参与课堂类图

2) 类间交互详细设计

通过上面对模块内类的说明分析了围棋资源与参与课堂时用户选择视频观看以及选择参与课堂时系统内类的设计。在此以用户参与课堂直播时的序列图说明系统内部的类间交互。如图 4-8 所示。

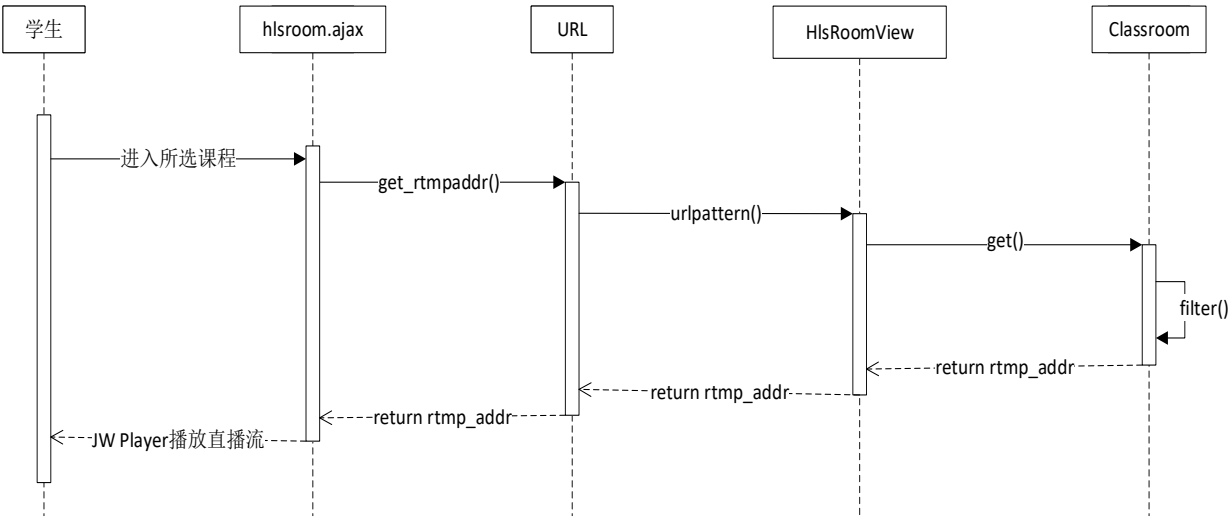


图 4-8 观看课堂直播时序图

学生用户拥有参与课堂的权限后,点击进入课堂,hlsroom.ajax 文件内 get\_rtmpaddr() 会被自动触发, django 的路由分发机制会将对应的请求送到对应的类中, 此处 HlsRoomView 处理课程的直播流问题。HlsRoomView 会根据前端提交的信息获取课程的 ID 信息, 在 Classroom 过滤后返回对应课程的 rtmpaddr 字段。最终 hlsroom.ajax 接收到直播流地址后, 通过 JW Player 可直接实现拉流播放。

## 4.2.4 校园管理与课程管理

### 1) 静态结构详细设计

校园管理与课程管理主要由“用户参与的校园与课程管理”和“教师维护的校园与课程管理”构成，其中用户参与的课堂即面向学生用户也面向教师用户。用户参与的校园管理与课程管理包括课堂的购买加入、信息查看和课堂删除，教师维护的校园管理与课程管理包括课堂信息的查看和更新以及课堂的删除，除此之外，校园管理还包括校内文件管理，班级管理还包括对参与课堂的学生的管理与统计。如图 4-9 为校园管理与课程管理的类图。

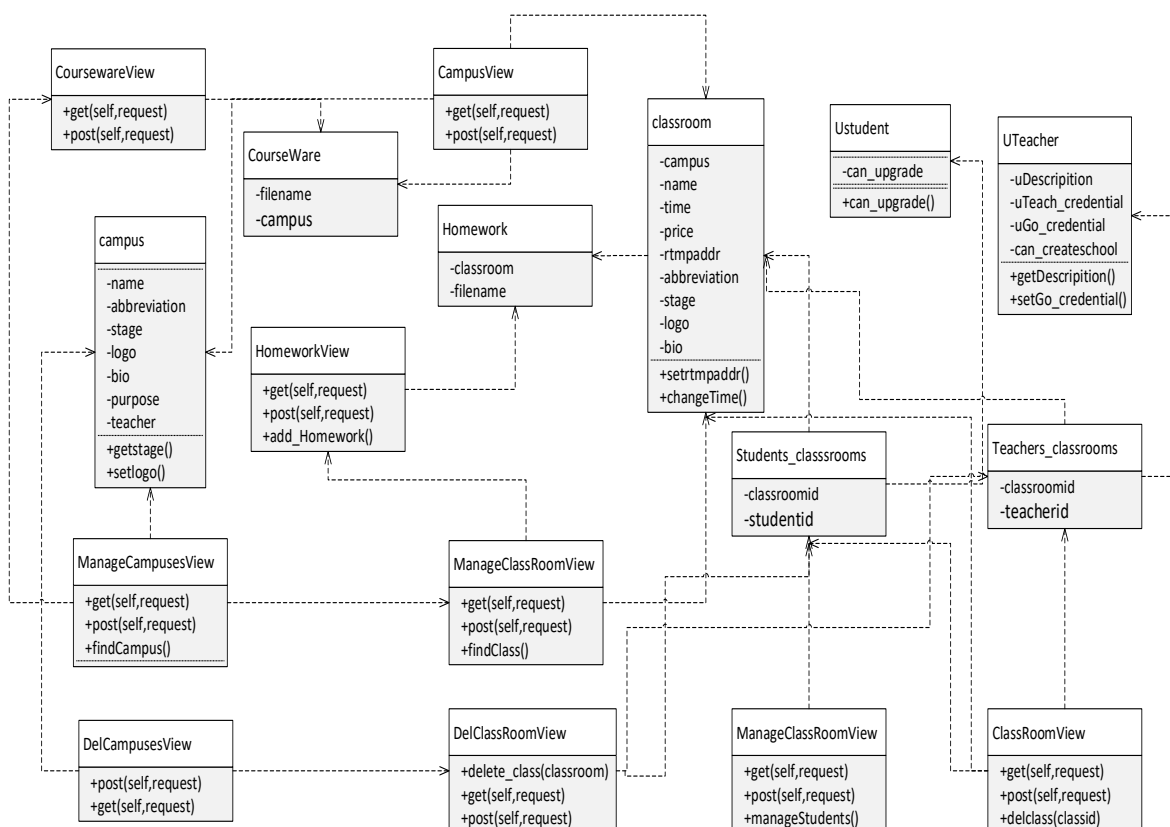


图 4-9 校园管理与课程管理类图

如图 4-9 所示, CampusView 和 ClassView 分别对应用户参与的学校与课程的管理, 包括信息查看与课程删除, 用户无权限删除学校, 在学校内无参与课程自动会从学校退出。用户对参与课堂的删除只是删除对应课表中的数据项, 即对 Teachers\_classrooms 或 Students\_classrooms 内某条数据的删除。ManageCampusView 和 ManageClassView 对应教师用户对所开课程的管理, 包括信息查看、信息更新。教师用户删除所开学校由 DelCampusView 受理, 删除课堂由 DelClassView 受理, 删除学校会自动删除校内所有课堂。删除课堂也会同时删除 Classroom、Teachers\_classrooms 以及 Students\_classrooms 中对应的数据项。校内文件管理由 CourseView 处理, HomeworkView 负责课堂内的文件管理。ManageStudentsView 通过 manageStudents()方法完成参与课堂学生管理, 主要是对学生人数的统计以及课堂破坏分子的删除。



## 2) 类间交互详细设计

通过上面对模块内类的说明分析了校园管理与课堂管理时系统内类的设计以及各类所负责的请求服务。在此以教师用户删除所开设课堂的序列图说明系统内部的类间交互，如图 4-10 所示。

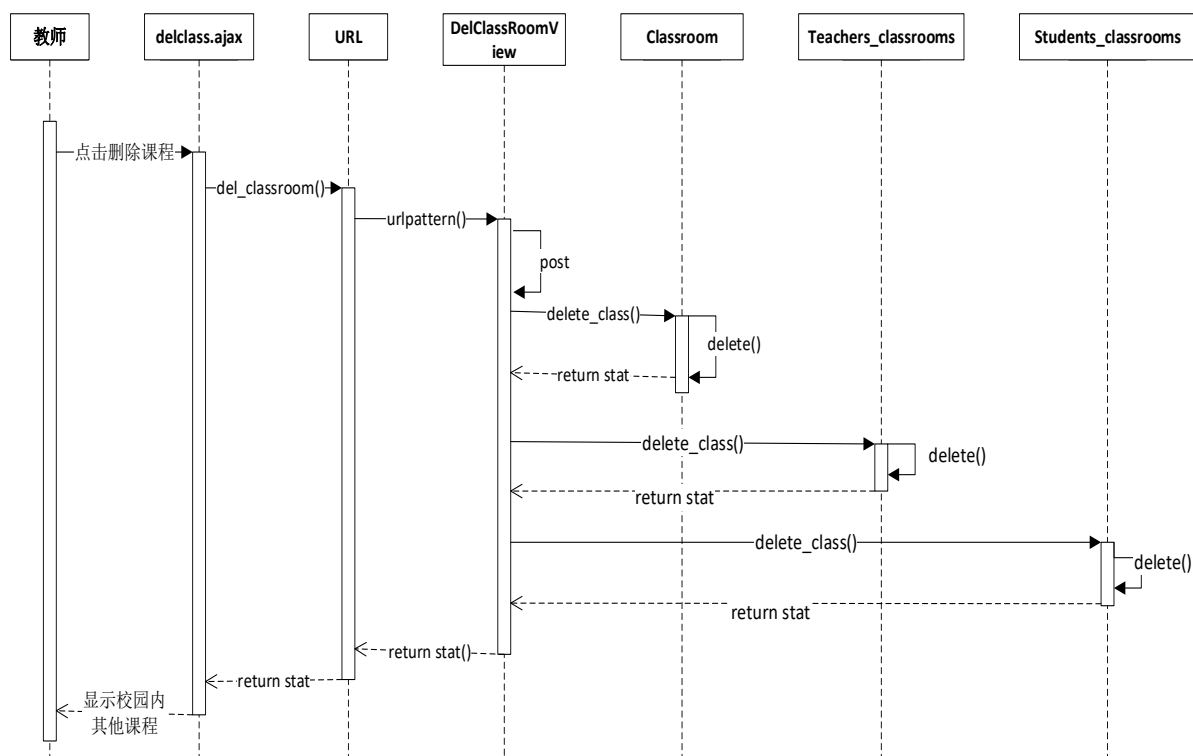


图 4-10 教师删除开设课程时序图

教师用户点击删除课程时，会触发校园页面的 `del_classroom()` 函数，向系统提交 post 请求。django 的路由分发机制会将对应的请求送到对应的类中，此处将由 `DelClassRoomView` 处理开设课程的删除问题。`DelClassView` 在 `post` 函数内通过 `delete_class()` 完成 `ClassRoom` 表内对应数据的删除，通过 `delete()` 方法返回删除是否成功的状态。`delete_class()` 函数还会通过判断依次删除 `Teachers_classrooms` 以及 `Students_classrooms` 中的数据。最终 `delclass.ajax` 文件内函数会做出判断，删除成功则不显示对应的课程信息，刷新页面只显示校园内的其他课程列表；删除失败会有“删除失败”的提示，并留在当前网页，不变化。

## 4.3 围棋直播教学系统的关键技术

### 4.3.1 利用 Nginx-RTMP Module 实现教学直播

Nginx RTMP，是 Nginx 托管在 Github 上的一个项目，其内部完全实现了 RTMP 协议，目的是可以使用户快速搭建起高效可实用的流媒体，即 NGINX-based Media Streaming Server<sup>[31]</sup>。Nginx-RTMP 虽然作为比较轻量级的项目，但其可以满足直播的采集，编码和分发工作，该项目丰富的完善了 Nginx 对视频的支持。作为 Nginx 的一个独立

的模块，在使用之前，需要下载 Nginx 和 Nginx-RTMP Module 的源码，并完成编译工作，之后 Nginx 便拥有了流媒体功能<sup>[32]</sup>。

通过对 Nginx 的配置，可以搭建起对围棋直播教学系统中直播课堂的底层支持。Nginx RTMP 提供了较多了流媒体服务，围棋直播教学系统中，分别实现了基于 RTMP 协议的直播服务，基于 HLS 协议的直播服务<sup>[33]</sup>，以及基于 RTMP 协议的点播服务<sup>[34]</sup>。系统中的参与课堂模块使用了基于 RTMP 协议的直播服务，学习资源模块中的围棋视频资源使用了基于 RTMP 协议的点播播服务。系统在流媒体服务器搭建中对 Nginx 的配置如下。

```
rtmp {
    #rtmp 流媒体服务器配置
    server {
        #rtmp 服务端口
        listen 1935;
        chunk_size 4096;
        #推拉流地址，例如：#rtmp://192.168.15.12:1935/live/{school_name}-{
        {classname}
        application live {
            live on;
            record all;
            record_path /home/loading_21th/LearningGo/;
        }
        #hls 协议的直播配置
        application hls {
            live on;
            hls on;
            #存储直播视频
            hls_path /tmp/hls;
            hls_nested on;
        }
        #点播配置，实现围棋视频资源的查看与播放
        application play {
            play /home/loading_21th/LearningGo/videos;
        }
    }
}
```

上述配置中 rtmp 关键字指明了流媒体服务器的协议，通过 server 分离了不同的流媒体服务器，listen 指明了监听端口，application 则标识了具体了流媒体服务实例<sup>[35]</sup>。本系统的流媒体服务实例为 live，具体地址为 rtmp://192.168.15.12:1935/live/{streamaname},streamname 可以作为更低等级的区分，例如可以采用不同的 streamname 实现围棋

直播教学系统中不同课堂之间的流地址隔离, `rtmp://192.168.15.12:1935/live/{streamaname}` 即是教师用户的推流地址, 也是学生用户的拉流地址。`record` 与 `record_path` 表示直播过程会被记录到指定地址<sup>[36]</sup>。

完成流媒体服务器的搭建后, 围棋直播教学系统系统中会根据学校简称以及课堂简称自动生成唯一的 RTMP 地址, 并反馈给教师用户。学生用户无权知道 RTMP 地址, 由浏览器自动获取并直接播放视频流。

### 4.3.2 利用 Websocket 与 Redis 实现课堂聊天

围棋直播教学系统为模拟直播课堂的真实性, 在直播过程中提供了课堂间分离的聊天室, 可以使教师用户与学生用户或学生用户彼此之间实时交流。直播课堂内部的聊天室采用了 Websocket 与 Redis 相关技术。

Websocket 作为 TCP 协议之上且实现了全双工长连接功能的新一代技术<sup>[37]</sup>, 用在聊天室中可以实现用户收发信息, 且不需要每次收发都消息建立新的连接, 并可以保证消息收发的容错率<sup>[38]</sup>。Redis 作为一种可以在内存中实现消息订阅发布的存储系统, 可以保证消息收发的快速有效性。Redis 需要设置相关频道, 并对同一频道的消息进行订阅和发布。课堂间分离的聊天室就是通过对不同课堂订阅不同频道的方法, 实现了聊天室彼此之间消息收发的隔离性。

围棋直播教学系统中, 前端网页通过 Websocket 请求与服务器建立连接后<sup>[40]</sup>, 即可实现消息的发送<sup>[39]</sup>, 系统获取网页中的学校简称和课堂简称作为 Redis 的消息频道, 进行订阅, 并将收取消息进行存储和发布。Redis 中新的消息发布后, 系统自动获取相关信息, 并发送到所有订阅此频道的网页, 即实现了隔离的互动聊天室。

## 4.4 围棋直播教学系统的数据库设计

上一节对围棋直播教学系统的四个功能模块通过类图和时序图进行了详细设计, 本节进行教学系统主要的数据库设计。数据库的设计直接决定着系统的实现方式, 系统的灵活性以及系统的功能。在本小节将对数据库的设计从两个方面进行论述: 第一, 数据库的概念模型设计, 主要搞清楚数据库中对应的实体及实体间的联系; 第二, 数据库的逻辑设计, 这部分将对数据库表给出详细的设计。

### 4.4.1 围棋直播教学系统的概念数据模型

本系统中的需要维护数据主要包括用户信息、学校信息以及课堂数据。具体包括学生用户, 教师用户的基本身份信息, 校园具体信息、校内开设课程以及校内围棋资源信息, 课程具体信息数据以及课程内的文件信息。除此之外, 系统还需为用户维护参与的课堂信息。系统维护这些数据需要通过规范的设计来降低冗余性。由于这些数据基本稳定以及表结构的合理性, 系统对这些数据的维护也只需要一些常规性的操作。本节通过描述系统的概念数据模型, 来展示系统中各类主要信息的实体以及实体之间的关联。系统的数据库 E-R 图如图 4-11 所示。

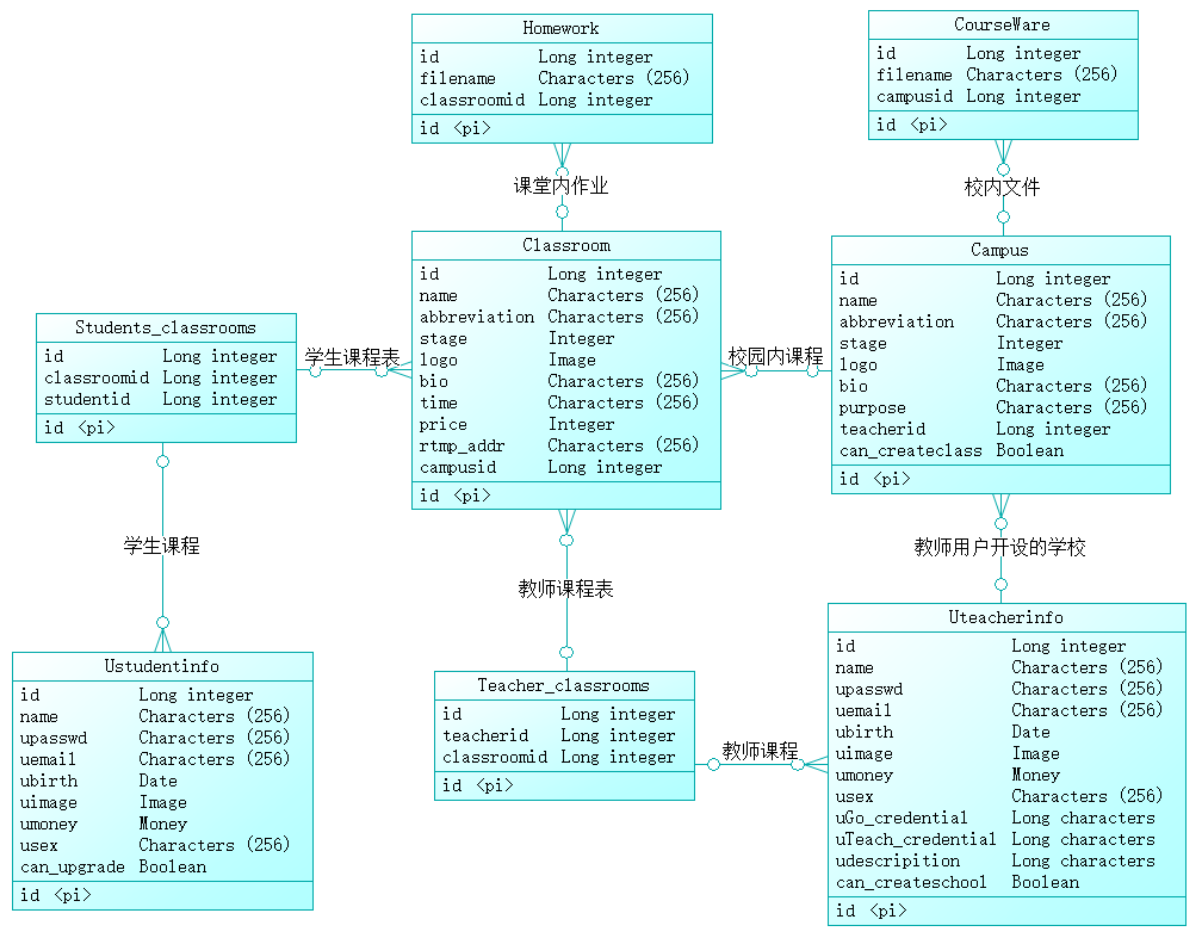


图 4-11 系统 E-R 图

#### 4.4.2 围棋直播教学系统的数据库表设计

上一小节对数据库的实体 E-R 图进行了展示，这一小节将对具体的数据库表进行详细设计，通过描述各表的具体字段以及字段限制，更清晰的展现系统内的数据结构。如下所示为数据库表详细设计说明：

##### 1) 学生用户信息表

表 4-1 学生用户基本信息表

字段名称	字段类型	主键/外键	含义
ID	Int	主键	学生身份标识
NAME	Varchar(256)		学生昵称
UWMAIL	Varchar(256)		邮箱地址
UBIRTH	Date		生日
UIMAGE	Varchar(256)		用户头像地址
USEX	Varchar(256)		性别
CAN_UPDATE	Int		是否有升级权限

学生用户的 CAN\_UPDATE 属性标识着用户是否购买了升级权限，付费后此属性为 1，可以升级为教师用户。

## 2) 教师用户基本信息表

表 4-2 教师用户基本信息表

字段名称	字段类型	主键/外键	含义
ID	Int	主键	教师身份标识
NAME	Varchar(256)		教师昵称
UWMAIL	Varchar(256)		邮箱地址
UBIRTH	Date		生日
UIMAGE	Varchar(256)		用户头像地址
USEX	Varchar(256)		性别
CAN_CREATE_SCHOOL	Int		是否有建校资格
UGO_CREDENTIAL	Varchar(256)		围棋段位水平证明图像地址
UTEACH_CREDENTIAL	Varchar(256)		教学水平证明图像地址
UDESCRIPTION	Text		教学描述

教师用户的 CAN\_CREATE\_SCHOOL 属性标识着用户是否购买了建校权限, 付费后此属性为 1, 可以创建新的学校。

## 3) 学校信息表

表 4-3 学校信息表

字段名称	字段类型	主键/外键	含义
ID	Int	主键	学校标识
NAME	Varchar(256)		学校名称
ABBREVIATION	Varchar(256)		学校英文简称
STAGE	Int		所属段位
LOGO	Varchar(256)		学校 LOGO 地址
BIO	Text		学校描述
PURPOSE	Text		办校宗旨
TEACHERID	Int	外键	创建教师 ID
CAN_CREATE_CLASS	Int		是否有开设课堂资格

学校的 CAN\_CREATE\_CLASS 属性标识着对应的教师用户是否购买了开课权限, 付费后此属性为 1, 可以在该学校内开设新的课程。

## 4) 课堂信息表

表 4-4 课堂信息表

字段名称	字段类型	主键/外键	含义
ID	Int	主键	课堂标识
NAME	Varchar(256)		课堂名称
ABBREVIATION	Varchar(256)		课堂英文简称

表 4-4 (续)

字段名称	字段类型	主键/外键	含义
RTMPADDR	Varchar(256)		课堂推流地址
CAMPUSID	Int	外键	所属学校 ID
STAGE	Int		所属段位
LOGO	Varchar(256)		课堂 LOGO 地址
BIO	Text		课堂描述
TIME	Text		开课时间说明
PRICE	Int		价格

课堂的 RTMPADDR 属性标识着该课堂的视频流地址，即直播过程中教师用户的推流地址和学生用户观看过程中的拉流地址。

#### 5) 视频信息表

表 4-5 视频信息表

字段名称	字段类型	主键/外键	含义
ID	Int	主键	视频标识
NAME	Varchar(256)		视频文件名称
STAGE	Int		所属段位
LOGO	Varchar(256)		视频图像地址
ADDR	Varchar(256)		视频文件地址

#### 6) 学生-参与课堂信息表

表 4-6 学生-参与课堂信息表

字段名称	字段类型	主键/外键	含义
STUDENTID	Int	主键	学生标识
CLASSROOMID	Int	外键	所属课程 ID

#### 7) 教师-参与课堂信息表

表 4-7 教师-参与课堂信息表

字段名称	字段类型	主键/外键	含义
TEACHERID	Int	主键	教师标识
CLASSROOMID	Int	外键	所属课程 ID

## 4.5 本章小结

本章对围棋在线教学系统进行了详细设计，包括：整体软件体系结构设计、功能模块层次化设计、对四个模块的详细设计以及数据库设计。其中，通过类图和时序图的方式对模块内部的类间关系和类间交互做了说明。这一章的详细设计将作为系统实现的输入，指导系统实现的完成。

## 5 基于 RTMP 流媒体技术的围棋直播教学系统的实现与测试

在上一章进行系统的整体软件结构设计和各功能模块的详细设计后，本章将着重于围棋直播教学系统的实现和测试。作为软件工程的落脚点，通过展示系统代码，运行截图和测试结果来更加清楚的展现系统的功能。

### 5.1 系统开发环境简介

本系统的开发环境采用 Ubuntu 系统，在一台物理机上完成整个开发过程，此物理机处理器为 64 位 Intel(R) Core(TM) 2 Quad CPU Q9400 @2.66GHz\*4，内存 8GB，硬盘 500G。系统开发的软件环境如表 5-1 所示。

表 5-1 系统开发软件环境

环境项	名称	版本
操作系统	Ubuntu (64 位)	16.04 LTS
开发语言	Python	3.5.2
开发框架	Django	1.9
数据库	MySQL	5.6
订阅发布服务器	Redis	3.0.6
反向代理服务器	Nginx	1.12.0
流媒体服务器	Nginx+RTMP	1.12.0
Web 业务逻辑服务器	uWSGI	2.0.15

### 5.2 围棋直播教学系统的实现

在上一节说明了系统的物理环境和软件的开发环境之后，本节将着重说明围棋直播教学系统的实现。系统的实现主要通过对围棋直播教学系统后端服务器的基本配置和各功能模块的代码来具体说明。

#### 5.2.1 后端服务器配置

##### 1) Nginx 配置

由于围棋直播教学系统基于直播实现，需要提供流媒体的服务，因此需要将普通的逻辑网络访问和推流、拉流的访问形式分开到不同的服务器。本系统采用 Nginx 作为反向代理服务器，接受用户的网络请求，然后依据不同的访问链接将不同的服务发送到 Nginx RTMP 配置的流媒体服务器，或者 uWSGI 业务逻辑服务器。Nginx 中基于 RTMP 实现的围棋直播教学配置已在 4.4 节介绍，在此不在赘述。

Nginx 不负责业务逻辑，具体的业务逻辑服务器由 uWSGI 完成，Nginx 通过 uWSGI\_params 向 uWSGI 转发请求。Nginx 与 UWSGI 的交互配置如下：

```

http {
    include      mime.types;
    default_type application/octet-stream;
    server {
        #业务逻辑服务器反向代理配置
        listen      8080;
        error_page  500 502 503 504  /50x.html;
        location = /50x.html {
            root    html;
        }
        location /LearingGo {
            #转发到 uWSGI 服务器
            include uWSGI_params;
            uWSGI_pass 192.168.15.12:9000; #python_uWSGI 服务器 ip:port
        }
    }
}

```

围棋直播教学系统利用 Nginx 本身的优越性作为静态资源服务器，配置如下。完成 css, JavaScript, 以及图片等静态文件的加载。

```

        location /static {
            alias /home/loading_21th/LearningGo/static;
        }
        location /favicon.ico {
            root /home/loading_21th/LearningGo/static/favicon.ico;
        }
    }
}

```

## 2) uWSGI 服务器配置

uWSGI 服务器负责围棋直播教学系统的逻辑业务功能，通过与 Django 框架的交互，完成用户发送 http 请求到接受系统服务的过程，uWSGI 服务器的配置如下：

[uWSGI]

```

#http = 127.0.01:9000 本地开发配置
socket = 192.168.15.12:9000 #通过 Socket 与 Nginx 交互
chdir = /home/loading_21th/LearningGo
module = LearningGo.wsgi
enable-threads = true #允许多线程
thunder-lock = true
ugreen = true #开启 python 协程，dwebsocket 模块使用
http-websockets = true #开启 websocket 服务
async = 100
master = true #主进程

```



```
processes = 9 #服务进程数量
```

```
vacuum = true
```

### 3) Django 框架的路由配置

Django 内部通过 URL 机制将不同的请求分发到不同的对象处理,作为最关键的控制器部分,开发人员只需要配置路由地址即可。本系统中最为重要的几条的路由配置如下:

```
url(r'^LearingGo/updateMoney/', WalletView.WalletView.as_view()) //充值与提现
url(r'^LearingGo/userinfo/addclass/(?P<campus_id>\d+)/', AddClassView.AddClassView.as_view()) //教师用户添加新课程
url(r'^LearingGo/hlsroom/(?P<campus_id>\d+)/(?P<class_id>\d+)/', HlsRoomView.HlsRoomView.as_view()) //直播页面展示
url(r'^LearingGo/userinfo/delclass/(?P<class_id>)', DelClassRoomView.DelClassRoomView.as_view()) //教师用户删除课程
...
```

## 5.2.2 功能模块的实现

### 1) 基本信息与虚拟钱包模块

基本信息与虚拟钱包模块中,系统主要维护学生用户和教师用户的基本信息和虚拟钱包的充值与提现。其中,用户充值与提现是其他主要模块的基础,通过充值用户虚拟货币增加,可以在系统内进行花费,教师用户在系统内盈利后可以提现。用户充值与提现的核心代码如下:

```
class WalletView(View):
    def updateMoney(user, money, is_add):
        nowmoney = user.umoney
        stat = "操作失败, 请重试"
        ...
        if (is_add == "yes"):
            nowmoney = nowmoney + int(money)
        else:
            nowmoney = nowmoney - int(money)
        ...
        if nowmoney < 0:
            stat = "提款失败, 余额不足"
        elif nowmoney > 10000:
            stat = "充值资金过多, 有风险"
        else:
            stat = "success"
            user.umoney = nowmoney
            user.save()
```

```

        return stat
    ...
    @csrf_exempt
    def post(self,request):
        """判断用户身份"""
        users = BaseTable.Uteacherinfo.objects.filter(id=request.session['uid'])
        if not users.exists():
            users = BaseTable.Ustudentinfo.objects.filter(id=request.session['uid'])
        if not users.exists():
            stat = 'fail'
        else:
            user = users[0]
            """充值/提现金额"""
            money_update_sum = request.POST.get('money_update_sum')
            """充值或提现"""
            is_add = request.POST.get('is_add')
            stat = updateMoney(user,money_update_sum)
        ...
        result = {'status':stat}
        ...
        return response

```

通过代码可知，用户充值或提现时，post 函数接受请求，解析充值或者提现的信息，取出变动金额，传递给 updateMoney 函数做处理。UpdateMoney 判断提现或充值是否合理，合理则更新数据库中用户的虚拟货币金额，不合理则返回失败消息。

## 2) 创办学校与开设课堂模块

课堂内教学是教师用户开展教学活动，学生用户进行围棋学习的最主要的途径。通过创办学校与开设课堂模块，系统为教师用户建立了可共享信息的校园，以及实现不同段位教学相隔离的直播课堂，方便了教师用户对于课堂的管理，学生也可以选择对应的合理课堂进入学习。在此，以教师用户开设新课堂的过程为例展示代码。

```

class AddClassView(View):
    def add_class(name,abbr,stage,bio,time,price,logo,rtmpaddr,campus):
        classroom = BaseTable.Classroom(name=name,
                                          abbreviation=abbr,
                                          stage=stage,
                                          bio=bio,
                                          time=time,
                                          price=price,
                                          logo=logo,
                                          rtmpaddr=rtmp,
                                          campus=campus)

```

```

classroom.save()
...
def update_campus(campus):
    campus.can_createclass=False
    campus.save()
@csrf_exempt
def post(self,request):
    campus = BaseTable.Campus.objects.get(id=request.POST['campusid'])
    """直播 rtmp 地址由系统根据学校简称和课堂简称生成"""
    rtmp="rtmp://192.168.12.105:1935/"+campus.abbreviation+"-"+request.POST.get
        ("abbr")
    """获取课堂信息"""
    name=request.POST.get('name')
    abbreviation=request.POST.get("abbr")
    stage=request.POST.get("stage")
    bio=request.POST.get("bio")
    time=request.POST.get("time")
    price=request.POST.get("price")
    logo=request.FILES.getlist("logo")[0],
    add_class(name,abbr,stage,bio,time,price,logo,rtmpaddr,campus)
    """课堂创建成功，学校的开课权限消失，下次开课需重新缴费"""
    update_campus(campus)
    """教师用户默认加入自己开设的课程"""
    teacher = BaseTable.Uteacherinfo.objects.get(id=request.session['uid'])
    teacher_classroom
BaseTable.Teachers_classes(teacher=teacher,classroom=classroom)
    teacher_classroom.save()
    ...
    """系统提示直播地址"""
    result = {'status':rtmp}
    ...
    return response

```

由上述代码可知，教师用户创建新课堂时，`post()`方法获取请求信息，解析出课堂的基本信息，传递给`add_class()`函数。其中，课堂的`rtmpaddr`属性，是由统一的URL与流名称结合组成，流名称由校园简称和课堂简称构成，由于学校与课堂的唯一性，从而实现了课程直播流的唯一性。`add_class()`方法通过`post`传入的信息实例化课堂对象，进行数据存储。新课堂开设成功后，系统会通过`update_campus()`方法更新对应校园的`can_createclass`属性，决定了一次购买只能创建一次课程的特性。最后，`post()`函数内部将教师用户加入了该课程，因此，教师用户对于所开课程拥有免费的观看权限。

### 3) 围棋资源与参与课堂模块

在线学校模块主要包含观看直播流, 实时互动聊天, 课件下载与作业上传等服务。围棋课堂的直播观看通过前端页面的 JW Player 到指定流地址拉流观看, 具体的流地址由系统通过检索对应的课堂, 返回其 `rtmpaddr` 属性。互动聊天采用了 Redis 的消息订阅发布机制, 为分隔不同课堂的消息内容, 采用学校名加课程名作为 Redis 的消息订阅发布频道, 同一课堂的学生教师订阅相同的频道。同样, 课件上传与作业下载时, 服务器端采用 `uploadfile` 目录下以学校名加课程名作为自路径存放对应的文件, 例如 XJTU 学校的 SOFT 课堂的所有文件路径在 `./uploadfile/XJTU/SOFT/` 路径下。下面通过作业上传的代码来做分析。

```
class HlsRoomView(View):
    def get(self,request,classroomid):
        """返回直播地址, JW Player 可直接播放"""
        classroom = BaseTable.Classroom.objects.get(id=classroomid)
        result = {'rtmp':classroom.rtmpaddr}
        """返回课堂内已有文件"""
        homeworks = BaseTable.Homework.objects.filter(classroom=classroom)
        result = {'homeworks':homeworks}
        ...
        return response
```

由代码可知, 用户进入课堂时, 系统通过 `get` 方法接受页面传送的课程 ID 信息, 系统通过 `Classroom` 对象检索对应结果, 获取对应课堂的 `rtmpaddr` 信息, 通过 `json` 返回。上述代码内还展示了进入课堂时, 系统会返回该课堂内的文件列表数据。文件的上传与下载由其他类实现。

### 4) 校园管理与课程管理模块

校园管理与课程管理模块主要分为参与课堂的用户对课堂的管理, 这类用户包括学生和教师, 只拥有对课堂的查看和删除自己所属课表的权限。校园管理与课程管理模块还包括教师用户对自己所开设课堂的增删改查的权限, 以及校园内的文件管理, 课堂内的学生管理。在此以教师用户删除所开设的课堂为例说明校园管理与课程管理模块的实现过程。

```
class DelClassRoomView(View):
    def delete_class(classroom):
        """获取参与该课程的教师课程表"""
        teacher_class = BaseTable.Teachers_classes.objects.filter(classroom=classroom)
        """获取参与该课程的学生课程表"""
        student_class = BaseTable.Students_classes.objects.filter(classroom=classroom)
        ...
        classroom.delete()
        teacher_class.delete()
```

```

        student_class.delete()
        return "success"
    ...
    def get(self,request,classroomid):
        """获取该课程"""
        classroom = BaseTable.Classroom.objects.get(id=classroomid)
        stat = delete_class(classroom)
        result = {'status':stat}
        ...
        return response

```

由上述代码可知, 点击删除课程按钮后, `get()`函数接收到请求, 通过 `classroomid` 等信息, 系统找出该课程, 传递给 `delete_class()`方法进行处理。`delete_class()`方法首先通过该课程信息, 检索出所参与此课程的教师用户和学生用户的课表项, 一一所删除, 之后, 对此课程信息完成删除, 返回状态信息。`get()`方法将此状态返回到前端页面。

## 5.3 围棋直播教学系统的测试

### 5.3.1 测试环境

本系统测试环境由客户端和服务端两部分组成, 服务端需包括系统运行所依赖的各种软件环境, 客户端主要包括链接 Internet 的浏览器。测试过程中, 在直播环境需要借助 `ffmpeg` 推流软件。具体测试环境如下。

#### 1) 客户端

硬件环境: 64 位 Intel(R) Core(TM) i5-6200U CPU Q9400 @2.30Ghz, 内存 8GB, 硬盘空间 256GB。

软件环境: 操作系统 Windows10(x64), 浏览器 Google Chrome 65.0.3325.146, `ffmpeg version N-89073-gff8f40a630 Copyright (c) 2000-2017 the FFmpeg developers.`

#### 2) 服务器端

硬件环境: 64 位 Intel(R) Core(TM) 2 Quad CPU Q9400 @2.66GHz\*4, 内存 8GB, 硬盘空间 500GB。

软件环境: Ubuntu 16.04 LTS (x64), Python 3.5.2, 数据库 MySQL 5.7, 反向代理服务器 Nginx 1.12.0, 订阅发布服务器 Rdeis 3.0.6, 流媒体服务器 Nginx+RTMP, 业务逻辑服务器 UWSGI 2.0.15。

### 5.3.2 功能性测试

在上一节客户端和服务端端的测试环境介绍完成之后, 本节将说明对围棋直播教学系统的功能性测试结果。功能测试是对系统中每个模块的各个功能点进行详细的测试。这一部分的工作主要采用的是黑盒测试技术, 黑盒测试着眼于程序外部结构, 在测试中, 把程序看作一个不能打开的黑盒子。因为针对每一个功能点都要经过详细测试,

所以这部分测试的工作量较大，系统功能测试的测试用例及结果如表 5-2 所示。

表 5-2 测试用例及结果

功能模块	功能点	期望结果	实际结果	结论
基本信息 与 虚拟钱包	用户注册	反馈注册结果	反馈注册结果	通过
	用户登陆	反馈登陆结果	反馈登陆结果	通过
	基本信息查看	返回基本信息	返回基本信息	通过
	基本信息更新	基本信息得到更新	基本信息得到更新	通过
	账户虚拟货币充值	账户虚拟货币增多	账户虚拟货币增多	通过
	账户虚拟货币提现	账户虚拟货币减少	账户虚拟货币减少	通过
	验证升级权限	是否有升级权限	是否有升级权限	通过
创建学校 与 开设课堂	购买升级资格	购买成功	购买成功	通过
	升级信息提交	升级成功	升级成功	通过
	验证建校权限	是否建校资格	是否建校资格	通过
	购买建校资格	购买成功	购买成功	通过
	验证教师用户开课权限	验证是否有开课可权限	验证是否有开课可权限	通过
	教师用户购买开课权限	购买成功	购买成功	通过
	教师用户提交开课信息	开课成功	开课成功	通过
围棋资源 与 参与课堂	选择段位寻找学校	根据段位选择学校	根据段位选择学校	通过
	查看学校信息和课程	返回学校基本信息和课程	返回学校基本信息和课程	通过
	参与课程验证	有无参与课程权限	有无参与课程权限	通过
	购买课程	购买课程成功	购买课程成功	通过
	收看课程直播	可以观看直播	可以观看直播	通过
	参与聊天互动	可以实时聊天	可以实时聊天	通过
	上传作业	上传作业成功	上传作业成功	通过
校园管理 与 课程管理	下载课件	下载课件成功	下载课件成功	通过
	查看参与课程	显示参与的课程信息	显示参与的课程信息	通过
	删除参与课程	删除参与课程，不再显示	删除参与课程，不再显示	通过
	学校文件上传	显示上传的文件	显示上传的文件	通过
	学校文件下载	下载指定文件	下载指定文件	通过
	课程文件上传	显示上传的文件	显示上传的文件	通过
	课程文件下载	下载指定文件	下载指定文件	通过
	教师用户更新开课信息	显示更新后的课程信息	显示更新后的课程信息	通过
	教师用户删除所开课程	相关课程全部删除	相关课程全部删除	通过
	统计课堂内学生人数	显示课堂内学生人数	显示课堂内学生人数	通过
	删除课堂内破坏分子	显示其余学生	未显示该破坏分子	通过

对比表 5-2 中的期望结果和实际测试结果，可知，围棋直播教学系统中基本信息与虚拟钱包、创建学校与开设课堂、围棋资源与参与课堂、校园管理与课堂管理四大功能模块的所有测试用例全部通过，较好的完成了系统的功能性测试。

### 5.3.3 非功能性测试

功能测试可以检测系统的可用性，非功能性测试则可检测系统的稳定性和易用性。非功能性的相关指标会直接影响用户的体验，为保证系统可以提供给用户稳定易用的服务，需要完成非功能性测试。由需求分析可知，围棋直播教学系统应主要在兼容性、直播时延、承受压力三个方面检测系统的稳定性。

1) 兼容性测试。围棋直播教学系统应满足用户在不同平台上参与围棋学习的要求,参与学习的用户只需要浏览器即可参与围棋课堂的学习。在不同平台、不同操作系统、不同浏览器下的测试结果如表 5-3 所示。

表 5-3 测试用例及结果

硬件平台	操作系统	浏览器	测试页面	结论
PC 机	Windows	谷歌浏览器	学校管理	通过
			课程直播	通过
		火狐浏览器	学校管理	通过
			课程直播	通过
	Ubuntu	Edge	学校管理	通过
			课程直播	通过
		谷歌浏览器	学校管理	通过
			课程直播	通过
iPad	iOS	Safari	学校管理	通过
			课程直播	通过
小米平板	Andriod	小米浏览器	学校管理	通过
			课程直播	通过
iPhone	iOS	Safari	学校管理	通过
			课程直播	通过
小米手机	Andriod	小米浏览器	学校管理	通过
			课程直播	通过

通过对表 5-3 中硬件平台和结论两项对比可知,围棋直播教学系统在手机、平板,PC 机等硬件环境下均支持良好,不同操作系统间系统的界面和服务均无明显差异,可以正常使用。系统提供正常的业务逻辑服务,对于课堂直播页面也提供了完善的支持,较好的完成了兼容性测试。用户在使用本系统过程中,也不需受限与不同的软硬件环境。

2) 直播课堂的时延测试。直播课堂的时延测试就是检测教师用户的视频采集与学生用户的视频观看在时间上的误差。过长时延直接影响用户体验,使教师讲解过程与学生观看直播时有一定的偏离。直播课堂的需求是保证教师用户的视频采集与学生用户的视频观看,应满足不长于 10s 的时延<sup>[41]</sup>。否则,课堂中教师用户与学生用户的交流会出现错乱,导致学生错过课堂中的围棋知识点。在此需求之下,需要完成直播课堂的时延测试,检测系统的时延误差。

在时延测试中,可在采用在电脑桌面放置走动的秒表,并通过 ffmpeg 软件的设置,将电脑桌面作为视频信息的采集点,并把桌面的信息推送到指定的流地址,即可将秒表的走动作为直播画面。在直播页面,当观察到桌面上走动的秒表后,与实际的秒表值作差,即可观察到直播的延迟是否在合理的范围之内。通过测试,时延误差的结果如图 5-1 所示,此结果为测试机与服务器在同一局域网内的测试结果。

由图 5-1 中的实际秒表和直播秒表对比可知,桌面上放置的秒表值为 00: 00: 20:

54, 将秒表值作为信息点推送到直播流地址, 浏览器页面拉取后, 播放时, 秒表信息值为 00: 00: 16: 34, 差值为 4 秒 20, 在合理的范围之内。较好的完成了直播课堂的时延测试, 4 秒 20 的时间延迟依然会对用户观看直播产生一定的影响, 但对于大多数的用户而言, 4 秒 20 应属于合理的误差范围。

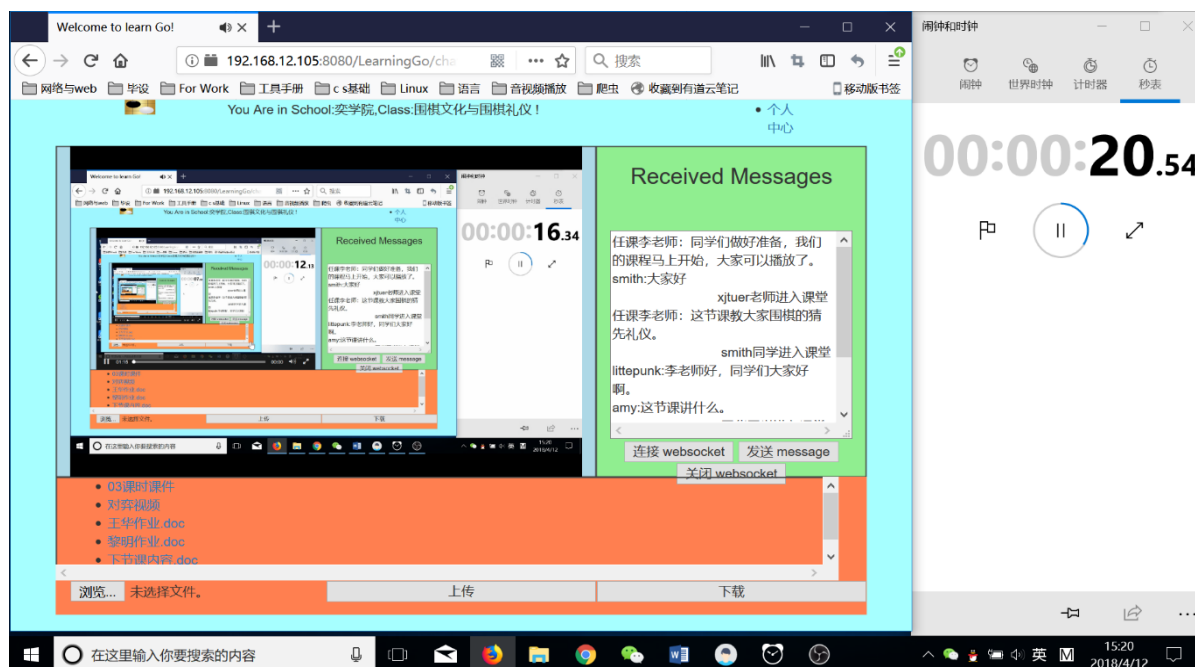


图 5-1 时延测试界面

3) 直播课堂的压力测试。直播课堂的压力测试是指, 围棋直播教学系统应在多个直播课堂同时进行, 每个直播课堂内有多名观看的学生用户的时候, 系统是否会崩溃。且压力测试过程中, 所有客户端均应该保持活跃状态, 与服务器正常连接, 并可以接收到视频数据。本文的压力测试, 课堂数为 10, 评价每节课设置 100 位学生用户拉流观看。

本次直播课堂的压力测试, 采用了 Linux 下开源的压力测试工具 `st_load`, `st_load` 需在 <https://github.com/rzrobert/st-load-master> 下载后, 进行编译, 编译完成即可安装使用。由于 `st_load` 工具内部支持 RTMP 流播放测试, 并且一个进程支持最高支持 5000 个并发的测试。`st_load` 会随时反馈所有客户端的拉流情况, 并在拉流结束后打印拉流的统计情况。Nginx-RTMP 模块也内置了 RTMP 视频流相关的统计功能, 并可通过网页观看视频流的统计情况, 具体配置如下。

```
server {
    listen 8081;
    location /stat {
        rtmp_stat all;
        rtmp_stat_stylesheet stat.xml;
    }
    location /stat.xml {
```



```

root /home/loading_21th/www;
}
location /control {
    rtmp_control all;
}
}

```

rtmp\_stat 和 rtmp\_stat\_stylesheet 为统计视频信息的命令。统计信息的查看可以通过访问/stat 地址来获取相关数据。

由于目前服务器仅在一台物理机上运行，因此对围棋直播教学系统的压力测试过程是通过对 10 个不同的课堂直播地址进行拉流，且每个课堂的 RTMP 地址设置 100 个拉流请求来进行。具体的单个测试命令如下：sb\_rtmp\_load -c 100 -r rtmp://192.168.12.105:1935/live/{StreamName}，对其中两个流地址的测试过程如图 5-2 所示。

```

tasks:0 estasks:0
[2018-05-09 16:00:28.578] [report] [16033] threads:100 alive:100 duration:3092 tduration:0 nread:0.00 nwrite:0.00 tasks:2078 etasks:1978 stasks:0 estasks:0
[2018-05-09 16:00:28.600] [report] [16011] threads:100 alive:100 duration:3092 tduration:0 nread:0.00 nwrite:0.00 tasks:2055 etasks:1955 stasks:0 estasks:0
[2018-05-09 16:00:28.643] [report] [16016] threads:100 alive:100 duration:3092 tduration:0 nread:0.00 nwrite:0.00 tasks:2170 etasks:2070 stasks:0 estasks:0
[2018-05-09 16:00:28.644] [report] [16029] threads:100 alive:100 duration:3092 tduration:0 nread:0.00 nwrite:0.00 tasks:1986 etasks:1886 stasks:0 estasks:0
[2018-05-09 16:00:28.771] [report] [16027] threads:100 alive:100 duration:3092 tduration:0 nread:0.00 nwrite:0.00 tasks:2101 etasks:2001 stasks:0 estasks:0
[2018-05-09 16:00:28.799] [report] [16020] threads:100 alive:100 duration:3092 tduration:0 nread:0.00 nwrite:0.00 tasks:1894 etasks:1794 stasks:0 estasks:0
[2018-05-09 16:00:28.811] [report] [16031] threads:100 alive:100 duration:3092 tduration:0 nread:0.00 nwrite:0.00 tasks:2124 etasks:2024 stasks:0 estasks:0
[2018-05-09 16:00:28.844] [report] [16026] threads:100 alive:100 duration:3092 tduration:0 nread:0.00 nwrite:0.00 tasks:2147 etasks:2047 stasks:0 estasks:0
[2018-05-09 16:00:28.905] [report] [16030] threads:100 alive:100 duration:3092 tduration:0 nread:0.00 nwrite:0.00 tasks:2101 etasks:2001 stasks:0 estasks:0
[2018-05-09 16:00:37.690] [report] [12466] threads:100 alive:100 duration:3542 tduration:0 nread:5.51 nwrite:0.00 tasks:100 etasks:0 stasks:0 estasks:0
[report] [12465] threads:100 alive:100 duration:3542 tduration:0 nread:12.72 nwrite:0.00 tasks:100 etasks:0 stasks:0 estasks:0
[2018-05-09 16:00:37.716] [report] [12464] threads:100 alive:100 duration:3542 tduration:0 nread:2.80 nwrite:0.00 tasks:100 etasks:0 stasks:0 estasks:0
[2018-05-09 16:00:37.755] [report] [12457] threads:100 alive:100 duration:3542 tduration:0 nread:1.00 nwrite:0.00 tasks:100 etasks:0 stasks:0 estasks:0
[2018-05-09 16:00:37.791] [report] [12462] threads:100 alive:100 duration:3542 tduration:0 nread:10.47 nwrite:0.00 tasks:100 etasks:0 stasks:0 estasks:0
[report] [12456] threads:100 alive:100 duration:3542 tduration:0 nread:0.55 nwrite:0.00 tasks:100 etasks:0 stasks:0 estasks:0
[2018-05-09 16:00:37.791] [report] [12463] threads:100 alive:100 duration:3542 tduration:0 nread:3.71 nwrite:0.00 tasks:100 etasks:0 stasks:0 estasks:0
[2018-05-09 16:00:37.795] [report] [12460] threads:100 alive:100 duration:3542 tduration:0 nread:1.90 nwrite:0.00 tasks:100 etasks:0 stasks:0 estasks:0
[2018-05-09 16:00:37.823] [report] [12450] threads:100 alive:100 duration:3542 tduration:0 nread:0.99 nwrite:0.00 tasks:100 etasks:0 stasks:0 estasks:0
[2018-05-09 16:00:38.003] [report] [12446] threads:100 alive:100 duration:3543 tduration:0 nread:0.55 nwrite:0.00 tasks:100 etasks:0 stasks:0 estasks:0
^C
loading 21thRTMP://st-load-master/objs

```

图 5-2 stload 工具压力测试界面

由图 5-2 中的 threads,alive 等各项数据指标可知，stload 对每个流地址采用 100 个线程同时拉流，其结果显示为各个拉流线程均保持活跃状态。在采用 stload 进行压力测试的同时，可在相应的统计页面观察到视频流的情况如图 5-3 所示。

RTMP	#clients	Video			Audio			In bytes	Out bytes	In bits/s	Out bits/s	State	Time
Accepted: 2961		codec	bits/s	size	fps	codec	bits/s	freq	chan				
live													
live streams	1000												
0	100		1.24 Mb/s			320 Kb/s				1.89 MB	41.55 MB	1.51 Mb/s	175.43 Mb/s
1	100		1.27 Mb/s			328 Kb/s				1.89 MB	41.97 MB	1.52 Mb/s	160.42 Mb/s
2	100		1.24 Mb/s			333 Kb/s				3.79 MB	231.51 MB	1.51 Mb/s	175.43 Mb/s
3	100		1.24 Mb/s			328 Kb/s				3.79 MB	228.08 MB	1.40 Mb/s	157.43 Mb/s
4	100		1.24 Mb/s			358 Kb/s				7.58 MB	610.44 MB	1.51 Mb/s	175.43 Mb/s
5	100		1.37 Mb/s			328 Kb/s				11.37 MB	988.6 MB	1.61 Mb/s	180.60 Mb/s
6	100		1.24 Mb/s			400 Kb/s				15.15 MB	1.336 GB	1.51 Mb/s	162.43 Mb/s
7	100		1.21 Mb/s			328 Kb/s				22.73 MB	2.075 GB	1.78 Mb/s	175.43 Mb/s
8	100		1.24 Mb/s			285 Kb/s				43.57 MB	4.113 GB	1.51 Mb/s	151.54 Mb/s
9	100		1.43 Mb/s			328 Kb/s				53.04 MB	5.035 GB	1.75 Mb/s	175.43 Mb/s
hls													
live streams	0												
play													
vod streams	0												

Generated by nginx-rtmp-module 1.1.4, nginx 1.12.0, pid 8954, built Jan 25 2018 15:18:09 gcc 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.5)

图 5-3 视频流信息统计界面

图 5-3 中, In Bytes 对应各 RTMP 地址的推流速率, Out Bytes 对应各 RTMP 地址的拉流速率, live stream 对应各流地址名称。可以观察到, 可知一共对 10 个直播课堂的 RTMP 地址进行了测试, 每个地址对应的拉流客户端都有 100 个。且各地址的推流速率在 1.51Mb/s 左右, 拉流速率在 175Mb/s 左右。在多个课堂的多个用户同时观看的情况下, 每个流地址均保持为活动状态 (State)。

## 5.4 围棋直播教学系统的运行截图

### 1) 基本信息与虚拟钱包

用户可随时进入个人中心查看和修改用户的基本信息, 包括修改昵称, 出生年月, 更换个人头像等。账户管理也在个人中心界面展示, 如图 5-4 为用户的个人中心界面, 模态框为用户充值页面。用户输入充值金额, 点击确认后系统会对充值请求做出反馈。

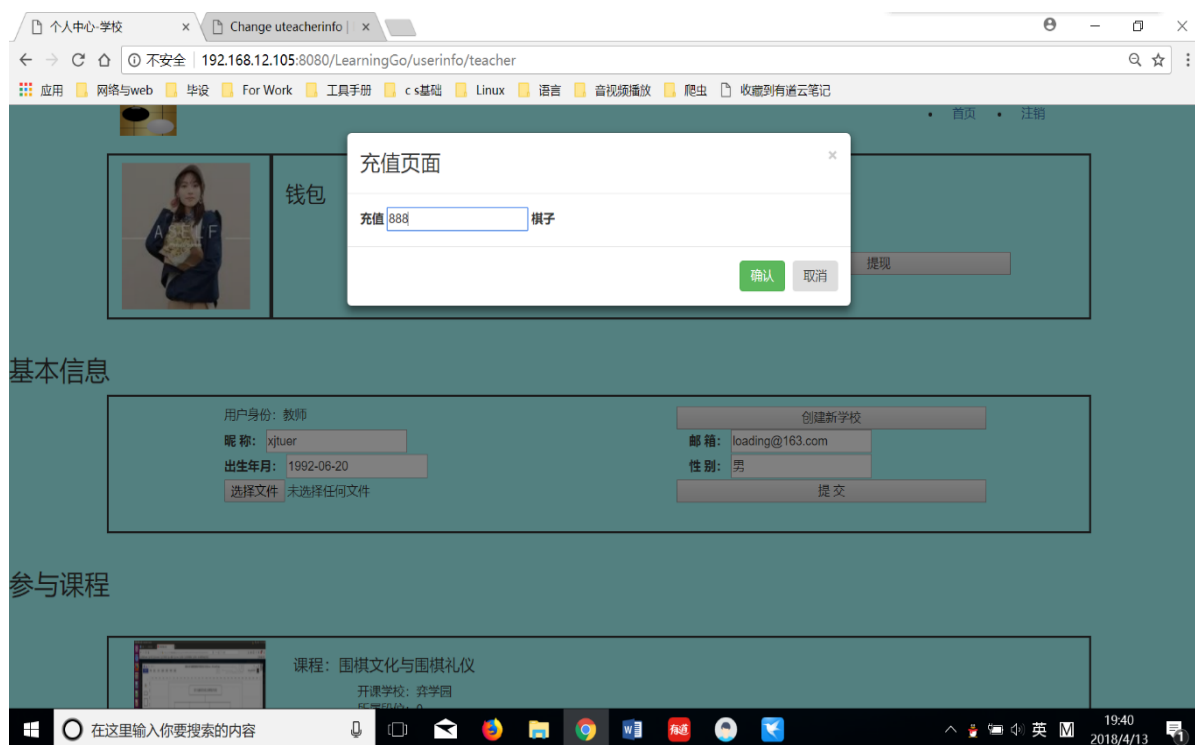


图 5-4 用户虚拟钱包充值界面

### 2) 创办学校与开设课堂

用户升级、创办学校与开设课堂均需要向系统支付费用, 完成支付后方可进行下一步的升级、建校以及开课。创建学校界面如图 5-5 所示。

页面中需要用户输入学校的学校名称, 学校简称, 学校所属段位, 教学宗旨, 学校描述等信息, 并在本地选择学校的 Logo 图片。点击确认之后, 系统会用相关信息实例化 Campus 对象, 并在系统数据库中查询是否已经存在的同名的学校, 没有相同的学校则会依据这些信息完成学校数据表的插入, 并存入数据库中。创学创建完成后, 对应教师用户的建校资格使用完成, 下次建校时会出现支付页面, 完成之后后拥

有建校资格。创建学校只对教师用户开发，学生用户界面不提供创建学校的选项和入口，学生用户无此权限。

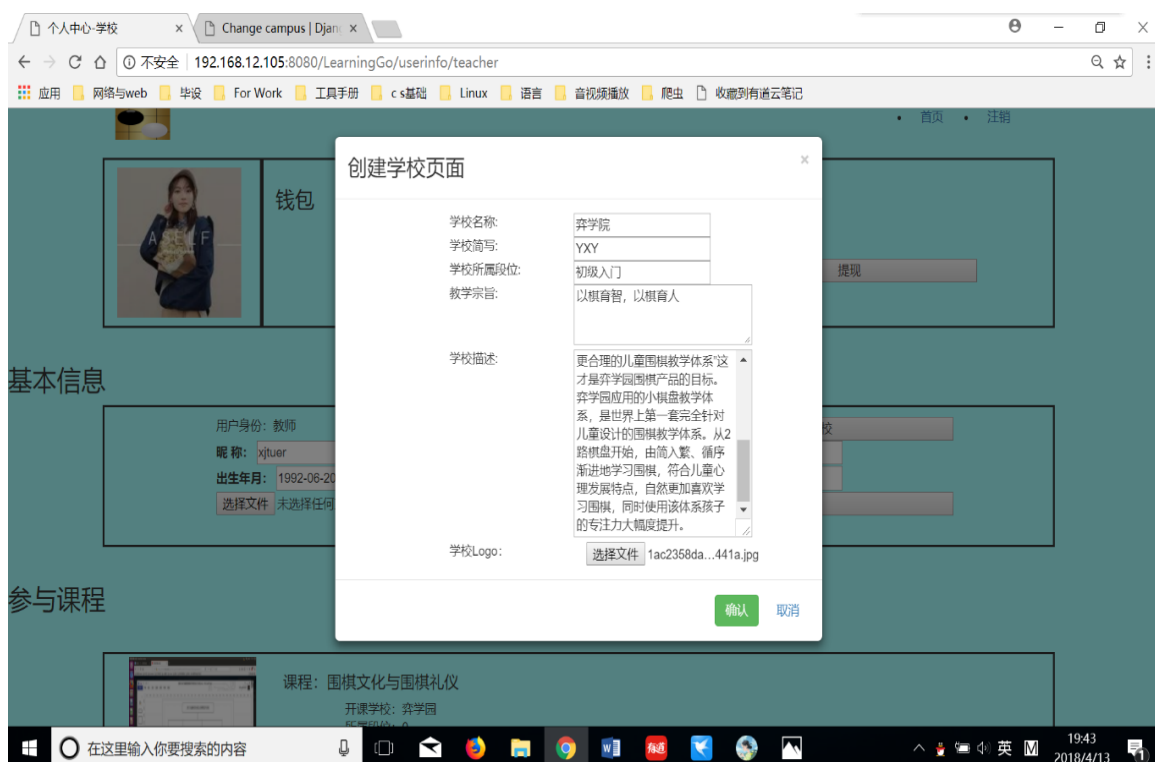


图 5-5 教师用户创建学校界面

开设课程与创建学校较为相似，如图 5-6 所示。课程开设界面，还需要教师用户填写开课时间，开课时间时教师用户与学生用户实现教学活动所必需的。除此之外，课程还需设置课程收费情况，学生用户缴费后才有参与课程的权限。

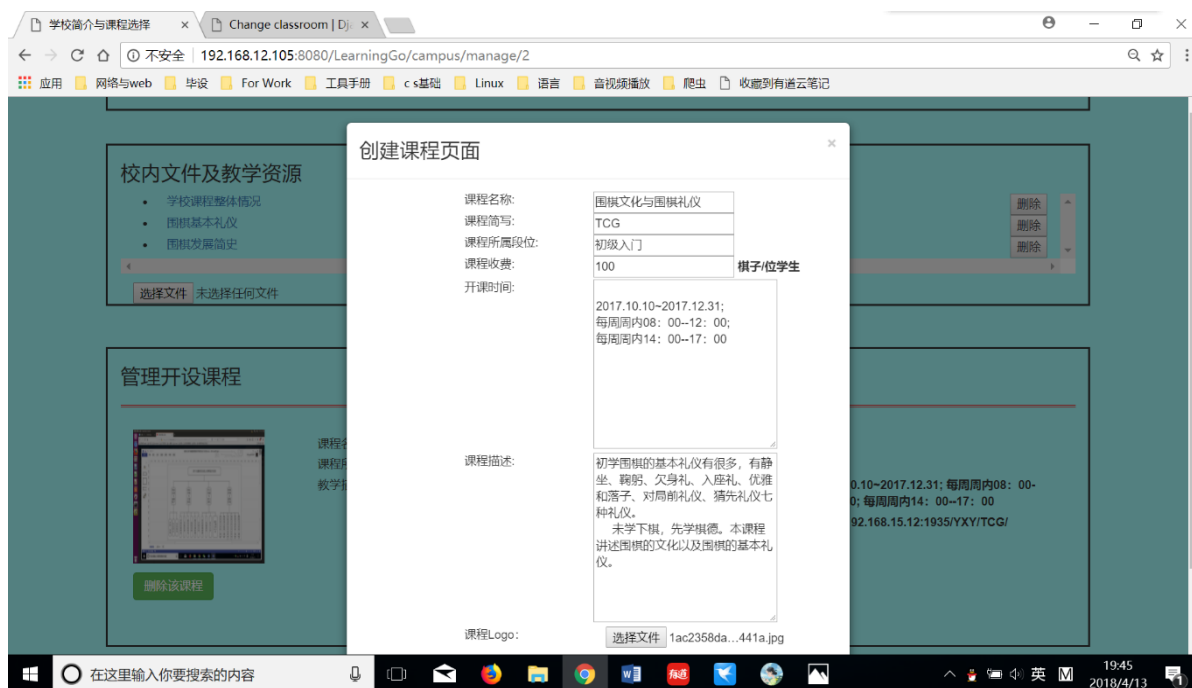


图 5-6 教师用户开课课程界面

### 3) 围棋资源与参与课堂

围棋资源与参与课堂是围棋直播教学系统开展教学活动主要形式。如图 5-7 所示为直播课堂的页面。



图 5-7 直播课堂界面

页面大致分为三块区域，直播界面，聊天界面，以及文件上传下载界面。用户通过直播界面观看教师直播教学，视频可最大化播放。聊天界面为用户提供了查看接收到的消息以及发送消息的小窗口，方便用户交流。文件上传下载区域，会显示该直播课堂内的所有文件，供用户下载查看。用户也可上传自己的文件到该区域。

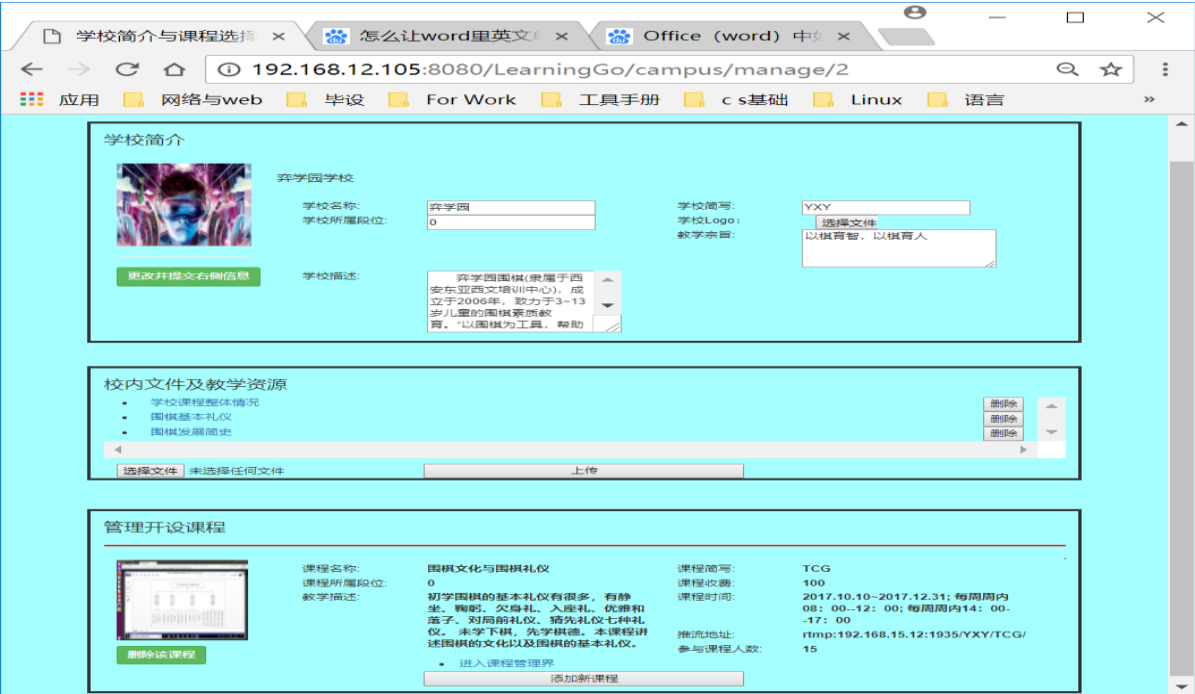
### 4) 校园管理与课程管理

课程管理分为参与学校和课程的管理与开设学校和课程的管理。参与课程管理的用户仅仅拥有查看学校和课程表中的学校和课程数据或从课程表中删除课程的权限，对课程信息无法修改。参与开设学校和课程管理的用户拥有对学校 and 课程相关数据的修改权限，且更新或删除课程信息时，用户的课程表中也会有相应更新。在此以学校管理界面为例，如图 5-8(a)所示为学学校管理界面。

由图 5-8(a)可知，官校管理界面提供了校园信息的查看和更新选项，教师用户可修改其中信息后提交。校园管理还包括对校园内文件的管理，教师用户可对其中的文件进行删除，下载或者上传新文件。校园管理界面还展示了校内的所有开设课程，管理开设课程部分展示了校内课程的信息，用户可以选择课程进入课程主页进行具体操作。在校园管理界面，用户可以删除其中的课程。

课程管理界面提供了课程信息的查看和更新选项，教师用户可修改其中信息后提交。课程管理还包括对课堂内文件的管理，教师用户可对其中的文件进行删除，下载或者上传新文件。课程管理界面还展示了参与该课程的所有学生用户信息，教师用户

对参与课程的用户拥有管理权限，对于捣乱分子，教师用户可以直接将其从课堂中剔除。最后，页面还展示了改课堂参与学生的人数，方便课程对学生进行统计，学校也可依此统计在校人数。具体的课程管理界面如图 5-8（b）。



(a) 学校管理界面



(b) 课程管理界面

图 5-8 校园管理和课程管理界面

## 5.5 本章小结

本章主要介绍说明了围棋直播教学系统的开发测试情况。通过对围棋直播教学系统的开发环境介绍，开发环境配置，功能模块的内部实现，测试环境的说明以及测试结果的描述，详细介绍了围棋直播教学系统的实现过程和测试过程，完整的展现了围棋直播教学系统的开发、运行、测试的情况。

## 6 结论与展望

### 6.1 结论

围棋运动作为一项益智类游戏对青少年智力的发展和稳重性格的养成有诸多好处,近年来围棋运动也越来越受到大众的关注和喜爱,也有越来越多的人投入时间和金钱去学习围棋。但由于线下围棋培训由许多局限性,基于 RTMP 流媒体技术的围棋直播教学系统的目的是将围棋的“教”“学”过程搬到互联网上,实现围棋教学资源的最大化利用。各线下的围棋培训学校也可以借助本系统内维护其教学资源,管理围棋课程及课堂内学生,并实现盈利。

围棋直播教学系统的主要工作可以概括为两点,其一通过 Django 框架建立了围棋的教学平台,支持教师用户开设学校和课堂,并对其进行管理;其二是建立了基于 RTMP 流媒体技术的直播课堂,使教师可以通过本系统对学生进行直播教学。

具体来说,本文主要工作内容包含以下几点:

1) 结合线下围棋培训的弊端,具体分析了围棋直播教学系统应该具有的主要功能。围棋直播教学系统为学生用户提供了免费的视频资源、可依据自身段位加入的围棋学校、覆盖从初级到高级段位的围棋课程;为教师用户提供了开设学校和课堂、管理学校及学校内围棋资源、管理课程及课堂内参与学生等功能模块,使围棋直播教学系统贴合围棋教学的实际需要。

2) 通过对 Django 框架进行研究和使用的,采用 CBV(Class-Based View)的方式搭建起围棋教学系统,对系统内基本信息与虚拟钱包、创办学校与开设课堂、围棋资源与参与课堂、校园管理与课程管理等功能模块一一进行实现。

3) 通过分析 Nginx-RTMP 模块,搭建起基于 RTMP 流媒体技术的直播系统,教师用户在获得系统分配的 RTMP 推流地址后,利用第三方软件将围棋讲解过程进行视频采集,并推流到该地址;学生用户在围棋课堂内,可直接在系统分配的页面拉流播放,完成围棋的直播教学。

4) 围棋直播教学系统还利用 Nginx-RTMP 实现了视频点播功能,并在围棋资源模块提供给用户多种视频资源,包括围棋名局赏析、经典棋局等。所有资源依据段位划分,极大的方便了用户的使用。

5) 围棋直播教学系统为实现更加真实的围棋教学环境,还通过 websocket 技术实现了课堂内部的互动。教师与学生进入直播课堂,即同时订阅了 Redis 的同一消息频道,系统利用 websocket 将 Redis 的消息实时同步到前端网页,对于围棋教学互动中的学生提问等环节进行了实现。



## 6.2 展望

由于围棋运动的推广和人工智能的发展，人们参与围棋学习的需求越来越旺盛，学习围棋的方式也越来越多样化。本文通过对当下围棋教学方式的分析，并结合互联网技术的发展，建设了一套具有可行性的围棋直播教学系统，为围棋教学提供了一种新的方式。但本文所做的工作由于时间和技术等方面的原因仍然非常有限，因此完成的功能也具有局限性。未来的研究工作主要集中在以下几个方面：

1) 加入双人对弈功能：本文只提供了在线课堂的学习形式，围棋学习的重点在于课下的实践，仅仅通过本系统的课程学习还无法牢固掌握围棋知识。在以后的工作中，应该加入在线的围棋对弈功能，使教师与学生或学生与学生之间加强围棋实践学习。

2) 加入人机对弈功能：学生通过本系统的自学方式只有观看视频，这一点十分具有局限性。以后的工作中，应加入人机对弈功能，对于课堂的学习学习可以在课后通过人机对弈进行消化吸收。

3) 加入人工智能参与对弈：本系统由于面向围棋学生和围棋教师，开展的功能以教学为主，缺乏对高端选手的围棋提升训练。以后的工作中，通过加入类似 AlphaGo 等人工智能的算法，开展高端选手与人工智能的对弈，将大幅提高人类的围棋水平。



## 致 谢

在本论文即将完成之际，谨向研究生阶段给与我帮助和指导的老师、同学和家人表示诚挚的感谢和深深的敬意！

首先，我要衷心感谢我的导师赵英良教授和指导老师赵银亮教授在三年研究生期间给予我许多的关怀与指导。他渊博的知识理论、严谨的治学态度以及勤勉的工作作风都让我深感敬佩。不仅仅在校学习期间做出了表率，也是本文以后的工作和生活中的榜样。从论文选题到最终完成，曾老师悉心指导并给出了非常宝贵和有建设性的意见。在此，谨向一直耐心指导本文的赵英良老师和赵银亮老师，表示诚挚的谢意！

其次，还要感谢软件学院的所有老师，感谢各位老师平时对我的悉心教导，为我传授了许多专业的知识，提供了良好的学习环境和科研条件。

再次，还要感谢各位同学和朋友在我生活和学习中给予的帮助和鼓励，我不会忘记和大家在一起的那些美好、欢乐的时光，祝福大家都拥有更美好的未来！

此外，还要感谢我的父母和家人多年来对我坚定不移的支持和无私的关怀，正是有他们多年的辛苦指导、鼓励和支持我才能顺利的完成学业。在论文完成之际，对他们致以最诚挚的谢意。

最后，感谢论文评审委员会的各位老师，在百忙之中抽出时间对本文进行评阅和指导。

## 参考文献

- [1] 李竹青. 西安地区围棋培训市场现状与发展研究[D].西安体育学院,2016.
- [2] 陈林. 基于 Django 的英语在线视频教学系统的设计与实现[D].厦门大学,2014.
- [3] 胡廷楣, 刘知青. 对面千里:人工智能和围棋文化[M]. 上海文化出版社, 2016.
- [4] 马勇.开展围棋学习活动 提高数学思维品质[J].甘肃教育,2018(03):52.
- [5] 马丛. 基于流媒体的音乐教学系统的设计与实现[D].电子科技大学,2012.
- [6] 2017 年运动训练、武术与民族传统专业招生新政出台[J].考试与招生,2017(01):7-9.
- [7] 葛会忠. 围棋改革已经在路上[N]. 中国体育报,2018-01-03(008).
- [8] 多九公. 欧洲围棋的播种者: 奥斯卡·科歇尔特[N]. 围棋报,2011-03-19(015).
- [9] 王利萍. 基于 Nginx 服务器集群负载均衡技术的研究与改进[D].山东大学,2015.
- [10] Li S, Si Z. Information Publishing System Based on the Framework of Django[M]// Advanced Graphic Communications and Media Technologies. 2017.
- [11] Li Z. Design and Implementation of the Software Testing Management System Based on Django[J]. Applied Mechanics & Materials, 2014, 525:707-710.
- [12] Gupta P, Zeldovich N, Madden S. A trigger-based middleware cache for ORMs[J]. Lecture Notes in Computer Science, 2011, 7049:329-349.
- [13] JEFF FORCIER. DJANGO WEB 开发指南[M]. 机械工业出版社, 2009.
- [14] 梁仕昌. 在线流媒体视频播放系统设计及实现[D].华南理工大学,2016.
- [15] 李波. 某互动教学直播系统的设计与实现[D].中国科学院大学(中国科学院工程管理与信息技术学院),2017.
- [16] 姜浩然,徐林.基于 RTMP 的流媒体服务器的研究[J].计算机与数字工程,2011,39(10):104-108.
- [17] Chang Q, Yang Z, Song Y. A Scalable Mobile Live Video Streaming System Based on RTMP and HTTP Transmissions[M]// Advanced Research on Computer Science and Information Engineering. Springer Berlin Heidelberg, 2011:113-118.
- [18] 孙盼,康维新.基于 DirectShow RTMP 协议的视频采集传输系统的设计与实现[J].应用科技,2015,42(06):40-44+76.
- [19] 陈木朝,姚超.RTMP 和 RTMFP 在视频播放系统中的应用比较[J].电脑编程技巧与维护,2013(10):82+87.
- [20] 张印. 基于 RTMP 的流媒体系统的设计实现[D].电子科技大学,2015.
- [21] 雷霄骅,姜秀华,王彩虹.基于 RTMP 协议的流媒体技术的原理与应用[J].中国传媒大学学报(自然科学版),2013,20(06):59-64.
- [22] 程承.基于 RTMP 协议的网络播出平台建设[J].科技创新与应用,2016(17):102.
- [23] 冯勋. 基于 RTMP 协议的轻量级视频服务器的设计与实现[D].电子科技大学,2015.
- [24] Lei X, Jiang X, Wang C. Design and implementation of streaming media processing software based on RTMP[C]// International Congress on Image and Signal Processing. IEEE, 2013:192-196
- [25] Bang Jun Cui. The Cross-Browser Multi-Platform Real Time Online Monitoring System Based on WebSocket[J]. Advanced Materials Research,2014,3470(1030).
- [26] 陆晨,冯向阳,苏厚勤.HTML5 WebSocket 握手协议的研究与实现[J].计算机应用与软件,2015,32(01):128-131+178.

- [27] 丁振凡. 基于 WebSocket 的在线围棋对弈软件设计[J]. 吉首大学学报(自然科学版), 2017, 38(04): 27-33.
- [28] 杨建业, 耿建平. 基于 HTML5 的实时 Web 数据监测系统[J]. 桂林电子科技大学学报, 2015, 35(02): 136-141.
- [29] 黄经赢. 基于 Socket.io+Node.js+Redis 构建高效即时通讯系统[J]. 现代计算机(专业版), 2014(19): 62-64+69.
- [30] 王放. 潜心黑白世界 凝神静静育人[N]. 长春日报, 2011-10-27(007)
- [31] Huang T, Li Z, Zhang J. Design and implementation of living streaming system based on multi-service nodes collaboration[C]// Advanced Information Technology, Electronic and Automation Control Conference. IEEE, 2017: 1635-16
- [32] 向华. 流媒体服务系统核心技术的研究与实现[D]. 西安电子科技大学, 2014.
- [33] LI Xin. Design of a Real Time Mobile Home Monitoring System[A]. 2013 年中国信息通信研究新进展论文集[C]., 2014: 5.
- [34] 孙业宝. 基于 RTMP 的高清流媒体直播点播封装技术的研究与实现[D]. 北京工业大学, 2015.
- [35] Liao Y, Wang Z, Luo Y. The design and implementation of a WebRTC based online video teaching system[C]// Advanced Information Management, Communicates, Electronic and Automation Control Conference. IEEE, 2017: 137-140.
- [36] 王艳. Android 系统中 RTMP 流媒体直播的设计与实现[J]. 电视技术, 2017, 41(01): 64-67.
- [37] Shen K, Si Z, Zhang L. Research and Implement of HTML5 Game Based on WebSocket[M]// Advanced Graphic Communications and Media Technologies. 2017: 423-428.
- [38] Acetozi J. Sending a User's Public Messages over WebSocket[J]. 2017.
- [39] 张艺. 基于 WebSocket 的即时通信系统研究与实现[J]. 软件, 2015, 36(03): 89-94.
- [40] 张玲, 张翠肖. WebSocket 服务器推送技术的研究[J]. 河北省科学院学报, 2014, 31(02): 49-53.
- [41] 宋西余. 基于 P2P 的流媒体直播系统的研究与实现[D]. 南京邮电大学, 2011.



## 学位论文独创性声明（1）

本人声明：所呈交的学位论文系在导师指导下本人独立完成的研究成果。文中依法引用他人的成果，均已做出明确标注或得到许可。论文内容未包含法律意义上已属于他人的任何形式的研究成果，也不包含本人已用于其他学位申请的论文或成果。

本人如违反上述声明，愿意承担以下责任和后果：

1. 交回学校授予的学位证书；
2. 学校可在相关媒体上对作者本人的行为进行通报；
3. 本人按照学校规定的方式，对因不当取得学位给学校造成的名誉损害，进行公开道歉。
4. 本人负责因论文成果不实产生的法律纠纷。

论文作者（签名）：                    日期：          年      月      日

## 学位论文独创性声明（2）

本人声明：研究生          所提交的本篇学位论文已经本人审阅，确系在本人指导下由该生独立完成的研究成果。

本人如违反上述声明，愿意承担以下责任和后果：

1. 学校可在相关媒体上对本人的失察行为进行通报；
2. 本人按照学校规定的方式，对因失察给学校造成的名誉损害，进行公开道歉。
3. 本人接受学校按照有关规定做出的任何处理。

指导教师（签名）：                    日期：          年      月      日

## 学位论文知识产权权属声明

本文声明，本文提交的学位论文及相关的职务作品，知识产权归属学校。学校享有以任何方式发表、复制、公开阅览、借阅以及申请专利等权利。学位论文作者离校后，或学位论文导师因故离校后，发表或使用学位论文或与该论文直接相关的学术论文或成果时，署名单位仍然为西安交通大学。

论文作者（签名）：                    日期：          年      月      日

指导教师（签名）：                    日期：          年      月      日

(本声明的版权归西安交通大学所有，未经许可，任何单位及任何个人不得擅自使用)