

硕士学位论文

**基于直播课堂的围棋教学系统的设计与实现**

学位申请人：李尧

指导教师：赵银亮教授

类别（领域）：工程硕士（软件工程）

2018年3月

**Design and Implementation of Go Teaching System Based on Live Classroom**

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: Yinliang Zhao

(Software Engineering)

March 2017

**论文题目：基于直播课堂的围棋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：赵银亮教授**

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围棋。但是，围棋的教学模式过于单一，导致围棋运动的推广极为有限，围棋爱好者也无法得到广泛的交流。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功能模型、静态模型、行为模型的分析，清晰的展现了在线围棋教学所需要包含的学生与教师用户的管理、在线的课堂学习、用户升级、创建学校、开设课堂、课堂管理等服务。在需求分析的基础之上，本文还说明了围棋教学系统的B/S架构以及系统整体的模块结构划分。通过具体分析各模块内部的静态结构以及实现在线互动等功能所需要的类间交互行为，使围棋教学系统的内部结构更加清晰，对用户提供的服务也更加明确。明确了需求和系统的内部结构，第五章完成了系统的实现和测试。实现包括开发的软硬件环境、服务器的配置和功能模块内的主要类与函数，测试包括测试环境以及主要测试用例的结果。

围棋教学系统在试验阶段和使用过程中，比较符合围棋的教学互动模式。用户基本信息的管理与学校和课堂的创建实现了“教”、“学”的分离以及用户的身份转换，在线学习实现了围棋教学的网上课堂，账户管理也为学校和教师带来盈利。

**关 键 词：**RTMP;流媒体视频直播；围棋

**论文类型：**工程设计

**Title:** **Design and Implementation of Go Teaching System Based on Live Classroom**

**Professional Fields: Software Engineering**

**Applicant: Yao Li**

**Supervisor: Prof. Yinliang 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 a 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.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 paper makes an analysis of Go teaching system based on Internet mode.First, by introducing Nginx, uWSGI and Django to illustrate the back-end server architecture of the teaching system, 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.Based on the requirement description and model, static model, the function of the system analysis of behavior model, clearly shows the online game teaching needs to include the students and teachers of the user management, online classroom learning, users to upgrade, and create a school, open classroom, classroom management, and other services.On the basis of demand analysis, this paper also explains the B/S architecture of go teaching system and the division of the module structure of the system as a whole.Through detailed analysis of the static structure of the module internal and online interactive functions such as interaction between classes, you need to go the internal structure of the teaching system more clear, service for users with more clear.The internal structure of requirements and system is clarified, and the implementation and testing of the system are completed in chapter 5.The implementation includes the development of the hardware and software environment, the configuration of the server, and the main classes and functions in the functional modules, including the test environment and the results of the main test cases.

In the experimental stage and the use of go teaching system, it is more in accordance with the teaching interactive mode of Go.The management of the basic information of the user with the school and class creation realized the separation of "teaching" and "learning" and the user's identity transformation, online learning has realized the online game teaching classroom, account management for schools and teachers also bring profit.

**KEY WORDS**: RTMP;Go;live streaming video

**TYPE OF THESIS**: Engineering Design

**目 录**

[1 绪论 1](#_Toc509261457)

[1.1 选题意义及应用背景 1](#_Toc509261458)

[1.2 国内外研究现状分析 2](#_Toc509261459)

[1.3 论文的主要研究内容 3](#_Toc509261460)

[1.4 论文的组织结构 4](#_Toc509261461)

[2 线上围棋教学系统的主要技术 5](#_Toc509261462)

[2.1 服务器架构与应用框架 5](#_Toc509261463)

[2.1.1 Nginx与反向代理 5](#_Toc509261464)

[2.1.2 Django与uWSGI 6](#_Toc509261465)

[2.2 流媒体技术与课堂直播 8](#_Toc509261466)

[2.2.1 直播架构 8](#_Toc509261467)

[2.2.2 RTMP协议与Nginx-RTMP模块 8](#_Toc509261468)

[2.2.3 JW Player 9](#_Toc509261469)

[2.3 课堂中的及时互动聊天 10](#_Toc509261470)

[2.3.1 基于Websocket实现长连接全双工通信 10](#_Toc509261471)

[2.3.2 Redis与聊天消息的订阅发布 11](#_Toc509261472)

[2.4 本章小结 12](#_Toc509261473)

[3 基于直播课堂的围棋教学系统的分析 13](#_Toc509261474)

[3.1 围棋教学系统的需求描述 13](#_Toc509261475)

[3.2 围棋教学系统的需求分析 14](#_Toc509261476)

[3.2.1 围棋教学系统的功能模型 14](#_Toc509261477)

[3.2.2 围棋教学系统的行为模型 20](#_Toc509261478)

[3.3 本章小结 26](#_Toc509261479)

[4 基于围棋的围棋教学系统的设计 27](#_Toc509261480)

[4.1 围棋教学系统的概要设计 27](#_Toc509261481)

[4.1.1 围棋教学系统的软件体系结构 27](#_Toc509261482)

[4.1.2 围棋教学系统的功能模块结构 28](#_Toc509261483)

[4.2 围棋教学系统的详细设计 29](#_Toc509261484)

[4.2.1 基本信息与账户管理 29](#_Toc509261485)

[4.2.2 在线学习 31](#_Toc509261486)

[4.2.3 用户升级与创办学校 33](#_Toc509261487)

[4.2.4 课堂管理 35](#_Toc509261488)

[4.3 围棋教学系统的数据库设计 37](#_Toc509261489)

[4.3.1 围棋教学系统的概念数据模型 37](#_Toc509261490)

[4.3.2 围棋教学系统的物理数据模型 38](#_Toc509261491)

[4.4 本章小结 41](#_Toc509261492)

[5 围棋教学系统的实现与测试 42](#_Toc509261493)

[5.1 系统开发环境简介 42](#_Toc509261494)

[5.2 围棋教学系统的实现 42](#_Toc509261495)

[5.2.1 后端服务器配置 42](#_Toc509261496)

[5.2.2 功能模块的实现 46](#_Toc509261497)

[5.3 围棋教学系统的测试 51](#_Toc509261498)

[5.3.1 测试环境 51](#_Toc509261499)

[5.3.2 功能性测试 52](#_Toc509261500)

[5.4 本章小结 53](#_Toc509261501)

[6 结论与展望 54](#_Toc509261502)

[6.1 结论 54](#_Toc509261503)

[6.2 展望 54](#_Toc509261504)

[致 谢 56](#_Toc509261505)

[参考文献 57](#_Toc509261506)

[附 录 58](#_Toc509261507)

[攻读学位期间取得的研究成果 59](#_Toc509261508)

可）：

CONTENTS

[1 Preface 1](#_Toc508962125)

[1.1 Significance and Background 1](#_Toc508962126)

[1.2 Glance of Current Research Status 2](#_Toc508962127)

[1.3 Main Research Contents of Thesis 3](#_Toc508962128)

[1.4 Organization of Thesis 4](#_Toc508962129)

[2 The main technology of online go teaching system 7](#_Toc508962130)

[2.1 Server architecture and application framework. 7](#_Toc508962131)

[2.1.1 Nginx 7](#_Toc508962132)

[2.1.2 Django与uWSGI 8](#_Toc508962133)

[2.2 Streaming media technology and classroom live. 9](#_Toc508962134)

[2.2.1 Broadcast architecture and RTMP 9](#_Toc508962135)

[2.2.2 Nginx-RTMP Module and JW Player 10](#_Toc508962136)

[2.3 Timely interactive chat in class 11](#_Toc508962137)

[2.3.1 long connection full duplex communication 11](#_Toc508962138)

[2.3.2 Redis and Subscription to chat messages 12](#_Toc508962139)

[2.5 The summary of this chapter 13](#_Toc508962140)

[3 The analysis of Go teaching system based on live broadcast 14](#_Toc508962141)

[3.1 Requirements description of go teaching system 14](#_Toc508962142)

[3.2 Demand analysis of go teaching system 15](#_Toc508962143)

[3.2.1 The function model of go teaching system 15](#_Toc508962144)

[3.2.2 The behavior model of go teaching system 21](#_Toc508962145)

[3.3 The summary of this chapter 27](#_Toc508962146)

[4 Design of Go teaching system based on Go 28](#_Toc508962147)

[4.1 A summary design of the go teaching system 28](#_Toc508962148)

[4.1.1 The software architecture of go teaching system 28](#_Toc508962149)

[4.1.2 The function module structure of go teaching system 29](#_Toc508962150)

[4.2 Detailed design of go teaching system 30](#_Toc508962151)

[4.2.1 Basic information and account management 30](#_Toc508962152)

[4.2.2 Online learning 32](#_Toc508962153)

[4.2.3 Users upgrade and start schools 34](#_Toc508962154)

[4.2.4 Classroom management 36](#_Toc508962155)

[4.3 Database design of go teaching system 38](#_Toc508962156)

[4.3.1 The concept data model of go teaching system 38](#_Toc508962157)

[4.3.2 Physical data model of go teaching system 39](#_Toc508962158)

[4.4 The summary of this chapter 42](#_Toc508962159)

[5 Implementation and test of go teaching system 43](#_Toc508962160)

[5.1 Introduction to system development environment 43](#_Toc508962161)

[5.2 The realization of go teaching system 43](#_Toc508962162)

[5.2.1 Backend server configuration 43](#_Toc508962163)

[5.2.2 Implementation of functional modules 47](#_Toc508962164)

[5.3 The test of the go teaching system 52](#_Toc508962165)

[5.3.1 The test environment 52](#_Toc508962166)

[5.3.2 Functional test 53](#_Toc508962167)

[5.4 The summary of this chapter 54](#_Toc508962168)

[6 Conclusions and Suggestions 55](#_Toc508962169)

[6.1 Conclusions 55](#_Toc508962170)

[6.2 Suggestions 55](#_Toc508962171)

[Acknowledgements 57](#_Toc508962172)

[References 58](#_Toc508962173)

[Appendices 59](#_Toc508962174)

[Achievements 60](#_Toc508962175)

可）：

# 绪论

## 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，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。

围棋作为中国传统文化的瑰宝，起源于春秋战国，发展于两晋南北朝，盛行于唐宋，明清时期也涌现出众多的围棋流派和名手。近年来，随着互联网和人工智能的发展，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。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。且由于在弈棋过程中，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。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，相应的其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，围棋培训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方面。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。线上的围棋教学系统，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；系统会审核开办学校的企业或个人的资质，拥有合理段位和教育经验的人方可通过审核，开办学校；同样，线上的围棋教学系统面向各个段位的选手，为围棋学子提供不同段位的视频资源和学校直播课程。线上的围棋教学系统可以大大节省资金和时间成本，使围棋教学者和围棋学生将更多的精力放在围棋教学和学习上。

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更加全方位和立体化，面向更多更高段位的围棋选手。同样，围棋的线上教学也有着潜在的巨大市场规模，完善的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。

## 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，目前国内外对于围棋的发展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》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，来自围棋行业、教育界、金融界、中科院的专业人士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，课件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。而且随着国内围棋爱好者和学习者的增多，线上围棋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。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围棋线上教学系统开发的初衷。

国内的线上围棋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；一些提供围棋对弈游戏平台的系统可以提供给围棋学习人员课余练习围棋对弈，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。

国内的线上围棋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，不受地域限制的与老师互动，有利于围棋教学资源的最大化利用。围棋线上系统应该学习这些教学系统的模式，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。

1. 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，其围棋教学尚未形成规模，线上围棋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到欢迎。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。日本的线上围棋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，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。

## 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教学系统。该系统面向围棋学生和围棋学校，学生和学校可通过注册登陆之后使用系统的免费视频浏览，管理课程，加入课堂学习和互动，创建学校与课堂，完成充值与提现等功能。其中，学生学习课程需要通过向学校支付虚拟货币，参与课程学习是通过围棋教师直播的形式，学习过程中可以参加与教师或与其他同学的互动。本论文具体的研究内容如下：

1）用户身份信息维护。围棋教学系统应区分两类用户，学生和教师。学生和教师初期采用同一形式的注册通道，注册成功后可持账号登陆进入网站学习，教师用户须通过升级模块进行转变。围棋教学系统应提供基本信息的维护管理模块，供用户更新头像和昵称等信息。学生和教师用户各自拥有唯一的虚拟账户，可进行虚拟货币的充值与提现，在相应支付页面进行花费。

2）在线学习。在线学习模块包含自学与加入课程学习两种方式。系统提供各段位的免费视频，用户通过选择感兴趣的段位标签，进入学习。学生与教师用户均可选择不同段位的学校和课程进行学习。加入学习的学校没有限制但在各学校内选择感兴趣的课程时，需要根据课程收费情况付费方可参与课堂。系统开展围棋教学的主要形式是通过教师直播，用户听讲的方式进行。教师通过OBS等软件推送相应的直播流到系统服务器，学生在相应网页即可观看，同时课堂中提供即时聊天和文件上传下载系统，使学生可与老师随时交流教学情况，完成课件下载或作业的上传。

3）用户升级与创办学校。普通学生用户可通过升级标签转为教师用户，只有教师用户拥有建校开课权限。升级与建校过程中需要向系统支付虚拟货币，上传相应的围棋教学资料，审核通过后，系统提示成功。

4）课程管理。用户在选择加入课程学习之前，需要向学校支付一定量的虚拟货币，虚拟货币流入对应的教师账户。学生可以查看参与的课程，添加或删除课程。教师用户在创办的学校内可开通新课程，须向系统完成支付，设置课程段位与开课时间，提交申请，系统会返回的相应的推流地址。教师用户对于所开设课程信息的更新与删除应同步所有用户参与的课堂。

## 论文的组织结构

论文从线上围棋教学的需求出发，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线上围棋教学系统的意义，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 线上围棋教学系统的主要技术。介绍开发线上围棋教学系统过程中所使用到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，直播的推流拉流相关的Nginx-RTMP模块，以及实现课堂内聊天室的websocket 和redis。

第3章 线上围棋教学系统的需求分析。对线上围棋教学系统进行需求描述，分析完成围棋教学所需要实现了各项功能。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完成的服务以及用户与系统的交互进行分析。

第4章 线上围棋教学系统的设计。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。本章根据系统的需求分析和详细设计，完成了系统的各个模块，以系统配置和模块内代码实现的形式论述了系统的个模块以及模块之间的交互和关联。最后，完成了系统的功能性的测试，通过测试用例说明系统是否符合需求。

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。

# 线上围棋教学系统的主要技术

本章主要介绍线上围棋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。

## 服务器架构与应用框架

### 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。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](https://baike.baidu.com/item/%E4%BB%A3%E7%90%86%E6%9C%8D%E5%8A%A1%E5%99%A8)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。线上围棋教学系统中，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。

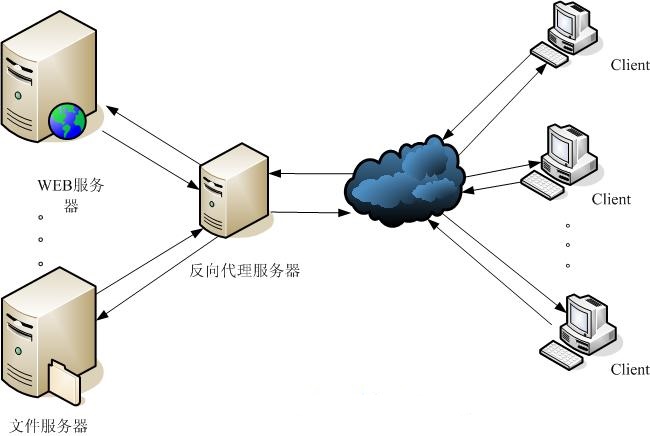


图2-1 反向代理服务器工作示意图

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。在搭建线上围棋教学教学系统的过程中，我们利用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。

### Django与uWSGI

#### 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。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。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-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，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。

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间交互以及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，我们利用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。

#### 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HttpUwsgiModule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方便的架构业余应用，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。

## 流媒体技术与课堂直播

### 直播架构

[流媒体](https://baike.baidu.com/item/%E6%B5%81%E5%AA%92%E4%BD%93)指将[音频](https://baike.baidu.com/item/%E9%9F%B3%E9%A2%91)、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。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](https://baike.baidu.com/item/%E5%8E%8B%E7%BC%A9/13032501)后存储到[流媒体服务器](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E6%9C%8D%E5%8A%A1%E5%99%A8)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。正是由于[流媒体技术](https://baike.baidu.com/item/%E6%B5%81%E5%AA%92%E4%BD%93%E6%8A%80%E6%9C%AF)的这种优越性，该技术广泛应用于[视频点播](https://baike.baidu.com/item/%E8%A7%86%E9%A2%91%E7%82%B9%E6%92%AD)、[视频会议](https://baike.baidu.com/item/%E8%A7%86%E9%A2%91%E4%BC%9A%E8%AE%AE)、远程教育、[远程医疗](https://baike.baidu.com/item/%E8%BF%9C%E7%A8%8B%E5%8C%BB%E7%96%97)和[在线直播](https://baike.baidu.com/item/%E5%9C%A8%E7%BA%BF%E7%9B%B4%E6%92%AD)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](https://baike.baidu.com/item/%E7%BC%93%E5%AD%98)、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](https://baike.baidu.com/item/%E5%AA%92%E4%BD%93%E6%9C%8D%E5%8A%A1%E5%99%A8)的性能和服务质量。因此，流媒体服务器是[流媒体](https://baike.baidu.com/item/%E6%B5%81%E5%AA%92%E4%BD%93)应用系统的基础，也是最主要的组成部分。在线上围棋教学系统中，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。

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OBS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，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

### 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](https://en.wikipedia.org/wiki/Macromedia)开发的[专有协议](https://en.wikipedia.org/wiki/Proprietary_protocol)，用于通过网络在[Flash](https://en.wikipedia.org/wiki/Adobe_Flash)播放器和服务器之间[传输](https://en.wikipedia.org/wiki/Streaming_media)音频，视频和数据。Macromedia现在归[Adobe所有](https://en.wikipedia.org/wiki/Adobe_Systems),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](https://baike.baidu.com/item/HTTP%E8%AF%B7%E6%B1%82)之上，可穿透[防火墙](https://baike.baidu.com/item/%E9%98%B2%E7%81%AB%E5%A2%99)；RTMPS类似RTMPT，增加了TLS/SSL的安全功能。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。在基于传输层协议完成三次“握手”之后后，RTMP协议也要客户端和服务器通过“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，这种格式的消息我们称之为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。

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，实现了堂课直播与缓存，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OBS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。

### JW Player

JW Player是一种基于flash的交互式网页媒体播放器。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Flah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，以便我们可以分享、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。

线上围棋教学系统采用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。

## 课堂中的及时互动聊天

### 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。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。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。而且它为我们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](https://en.wikipedia.org/wiki/Port_(computer_networking))号80（或在[TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security)加密连接的情况下为443 ）完成的，这对于那些使用[防火墙](https://en.wikipedia.org/wiki/Firewall_(computing))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2-3所示。

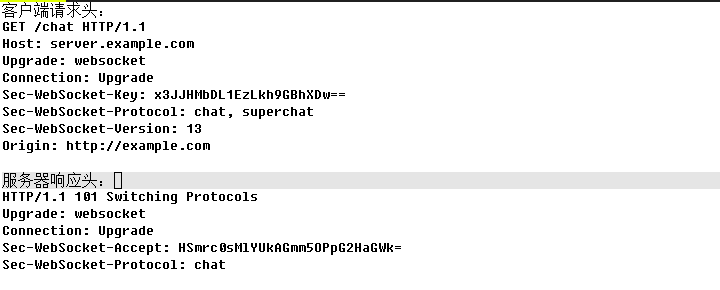


图2-4 websocket的请求头和响应头

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，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@qccept\_websocket装饰器即可。

### 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。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2-4，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

在线上围棋教学系统中，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。

## 本章小结

本章主要介绍了实现围棋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。

# 基于直播课堂的围棋教学系统的分析

上一章对本系统开发所涉及的主要技术进行了介绍，本章将对围棋教学系统进行需求分析。将以围棋教学面临的问题为切入点描述系统的业务需求，确立解决各主要需求所要提供的教学服务，阐述服务过程中的用户角色。本章将根据不同的用户类别研究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。

## 围棋教学系统的需求描述

线上围棋教学系统，要解决的问题是将包括教师和教学技巧在在内的围棋教学资源进行整合，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。线上围棋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，学生参与直播课堂与升级为教师用户同样需要缴纳费用。线上围棋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：

#### 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。

#### 虚拟货币账户管理需求描述

盈利同样是围棋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、收费学习的方式。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。

#### 课堂在线学习需求描述

围棋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。

#### 用户升级管理与创办学校需求描述

围棋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，学生用户需要升级才可拥有创办学校的权力，且升级过程中需要提交相关围棋水平或教学资质的证明，系统审核通过后方可拥有创立学校的资格。用户升级后，在创办学校过程中，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。学生用户升级以及教师用户创办学校，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。

#### 学生与教师的课堂管理需求描述

学生与教师均可参与课堂，参与的课堂可以有多项。学校也可针对不同段位开设不同的课程。系统需要对不同用户的多项课程进行管理和维护。学生用户新加入课程，需要对学校支付虚拟货币，货币流向学校所属的教师账户。教师用户开设新的课堂也要支付虚拟货币，货币流向系统账户。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。用户同样可以退出选择的课堂。教师用户默认对自己开设的课堂有参与观看的权限。教师用户需维护参与的课程与开设的课程。对所开设的课程信息的更新与删除应在所有对应的课表中进行更新。

## 围棋教学系统的需求分析

首先根据系统的功能性需求，利用用例图为系统建立功能模型；然后通过类图描述系统的内部信息结构和内部行为，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，建立系统的行为模型。

### 围棋教学系统的功能模型

#### 系统用例分析

本系统是一款线上的围棋教学系统，用户通过本系统可以学习围棋知识、开通围棋学校或通过授课的方式获取经济利润。使用者通过参与围棋课堂可以直接观看教学直播，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、课堂学习中文件数据维护、用户课程表管理等作用。由此可知，围棋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。

针对需求描述中的四类主要需求，围棋教学系统可通过基本信息与账户管理、在线课堂学习、用户升级与创办学校、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

基本信息与账户管理中，学生和学校完成注册、登陆、修改信息、花费虚拟货币、充值与提现等功能，系统管理员负责维护和管理各类信息；在线学习需要学校开通课程、教师用户直播和上传课件、学生用户观看和上传作业、共同参与讨论等，系统管理员负责维护管理上传下载的课件与作业。在线学习还应提供不同段位的学校划分，不同段位水准 的用户可选择进入合适的学校和课程进行学习；用户升级与创办学校服务功能中，需要学生用户提出申请，系统管理员审核，升级成功后，创办学校由教师用户完成，其间的用户升级与创办学校，需要收取费用获取相应的资格；学生和教师的课程管理功能，需要提供学生参与课程，教师参与课程，教师开设课程等信息的基本管理项目，这些信息的维护同样需要系统管理员在系统后台的参与。

#### 基本信息与账户管理用例分析

用户基本信息与虚拟账户管理是围棋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各功能模块中的数据。如图3-2所示为基本信息与账户管理用例图，其中包括了用户的基本信息管理和账户管理两大部分。

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，可以在登陆系统后进入用户信息页面进行查看自己的基本信息，也可以随时更改基本信息，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。

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，用户可以在登陆系统后进入基本信息页面查看自己的账户余额，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教学系统内部基本的虚拟货币功能。



图3-2 基本信息与账户管理用例图

用户基本信息修改不区分学生用户和教师用户，是用户保持自己基本信息为最新状态或最合适状态的管理服务。此处以用户基本信息修改为例做详细的用例描述，如表3-1所示。

表3-1 用户信息修改用例描述

|  |  |
| --- | --- |
| 用例名称 | 用户信息修改 |
| 简要说明 | 用户修改昵称，生日等信息 |
| 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，页面显示新的用户信息 |
| 基本流程 | 1.进入用户信息界面；  2.修改对应选项的用户信息；  3.点击提交，保存修改内容。  4.系统提示修改成功。 |
| 扩展流程 | 修改信息中出现非法字符，提升修改失败。 |

#### 在线学习用例分析

在线课堂学习是围棋教学系统最为核心的需求之一，在线用户主要面向学生用户和教师用户，后台管理员负责维护各功能模块中的数据。如图3-3所示为在线学习用例图。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，系统提供了单机围棋游戏，用户登陆系统后，可进入游戏页面，与朋友进行围棋游戏；自学模块中，系统还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。



图3-3 在线课堂学习用例图

加入课堂学习是在线学习最为主要的功能。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，推流地址在课堂管理模块中指定。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，教学课堂互动的用例描述如表3-2。课堂学习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。

表3-2 课堂教学互动用例描述

|  |  |
| --- | --- |
| 用例名称 | 课堂中教学互动 |
| 简要说明 | 上课过程中，学生与学生之间，教师与教师之间，可以发言交流，增加互动 |
| 执行者 | 学生或者教师用户 |
| 前置条件 | 用户已经登录系统 |
| 后置条件 | 发言内容显示到互动交流区，同时可观看其他同课堂用户的发言内容 |
| 基本流程 | 1. 选择学校和课程，进入教学直播界面； 2. 在互动交流区阅读别人发言； 3. 在输入框中输入发言文字； 4. 点击提交，参与互动。 |
| 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 |

#### 用户升级与创办学校用例分析

用户升级与创办学校作为围棋教学系统最为核心的需求之一，主要面向学生用户、教师用户和后台管理员，后台管理员负责维护各功能模块中的数据。如图3-4所示为用户升级与创办学校用例图。



图3-4 用户升级与创办学校用例图

用户升级主要针对学生用户，用户升级后成为教师用户，拥有创办学校权限；创办学校针对教师用户，教师用户可以创办学校；用户升级和创办学校均需要购买相应资格。用户获取升级资格后，应重新提出申请，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，之后登陆系统自动识别为教师用户。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，之后教师登陆系统会自动加载所创建的学校。

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。具体的学生用户升级用例描述如表3-3所示。

创建模块中，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。教师用户在之后的开设课堂环节中，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。开设课堂需购买开课资格，每所学校对应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。

表3-3 学生用户升级用例描述

|  |  |
| --- | --- |
| 用例名称 | 学生用户升级 |
| 简要说明 | 学生用户升级，升级之后成为教师用户 |
| 执行者 | 学生 |
| 前置条件 | 用户已经登录系统 |
| 后置条件 | 系统自动识别为教师用户 |
| 基本流程 | 1. 进入用户信息界面； 2. 点击用户升级按钮； 3. 若无升级资格，系统会自动进入支付页面，花费指定数量的虚拟货币购买升级资格； 4. 在用户升级页面输入升级所需资料； 5. 点击提交，等待审核； 6. 后台管理员进入审核页面，检查相关资历； 7. 检查通过后选择审核通过； 8. 系统显示用户为教师用户。 |
| 扩展流程 | 购买创办资格需要扣除虚拟货币，虚拟货币不足无法拥有创办资格； |

#### 课程管理用例分析

课程管理作为围棋教学系统最为核心的需求之一，主要面向学生用户、教师用户和后台管理员，后台管理员负责维护各功能模块中的数据。如图3-5所示为课程管理用例图。课程管理模块主要包括两类用户对参与课程的管理，包括添加、查看和删除，教师用户对开设课程的管理，包括新增、更新、查看与删除。



图3-5 课程管理用例图

用户加入课程前需要首先购买参与课堂的资格，课堂费用由教师开设课堂时设定，用户花费指定数量的虚拟货币，虚拟货币进入教师账户，完成支付，获取参与课堂的资格之后，学生可以加入课堂。如表3-4即为学生用户新添加课堂的用例描述。

表3-4 学生用户添加课程用例描述

|  |  |
| --- | --- |
| 用例名称 | 学生用户添加课程 |
| 简要说明 | 学生用户选择感兴趣课程，参与直播学习 |
| 执行者 | 学生 |
| 前置条件 | 用户已经登陆系统 |
| 后置条件 | 学生拥有参与课堂的权限，可以观看直播，参与讨论，下载课件，上传作业 |
| 基本流程 | 1. 在系统中选择合适段位； 2. 选择推荐学校； 3. 进入学校； 4. 点击‘参与/购买 课程’； 5. 用户拥有课程参与权限，直接进入课堂，否则，进入支付页面，花费指定虚拟货币，完成支付； 6. 参与课堂学习，可以观看直播，参与互动，下载课件，上传作业。 |
| 扩展流程 | 学生拥有的虚拟货币不足，无法完成课程购买，添加课程失败 |

开设课堂需要教师用户首先购买开设课堂的资格，开设课堂所需费用由系统制定，教师用户花费指定数量的虚拟货币，虚拟货币从教师账户中扣除，完成支付。获取开设课堂资格后，教师用户可以设立课程信息并发布，设立课程信息需要提交，提交之后系统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OBS等软件将直播流推送到指定地址即可完成课堂直播。

课程信息维护面向学生和教师用户，包括课程信息查看，更改课程信息，删除课程等内容。学生可以通过课程信息查看获取课程直播时间、课程简介等内容。教师用户对所开设的课程还可以获取推流地址等信息；教师用户更改课程信息会同步到所有参与此课程的用户的课程表中，教师对所开课程的删除也会在学生课表中完成删除。

### 围棋教学系统的结构模型

结构模型是探索问题领域的工具，可以帮助我们探索和提炼问题领域知识，它是对实际问题空间的抽象而非解空间。在对系统进行业务建模时主要提取系统业务相关的关键词，或者领域专业名词中提取出顶层类。面向热点话题发现的Web新闻分析系统在问题空间进行抽象的静态模型如图3-3所示。下面将对类做进一步地解释。

### 围棋教学系统的行为模型

通过对围棋教学系统功能模型的分析系统总体分为基本信息与账户管理、在线课堂学习、用户升级与创办学校和课程管理四个功能模块，下面分别具体描述这些功能的行为模型。

#### 基本信息与账户管理行为模型

基本信息与账户管理是指学生用户和教师用户在使用系统的过程中可以拥有并维护用户身份的基本信息和账户，账户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

在用户账户充值活动图中，操作对象包括学生用户和教师用户。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。

#### 在线学习行为模型

在线学习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和参与课堂的教学式学习。参与课堂式学习过程中，用户需要采用观看直播的形式，在此以观看直播为例描述在线学习的行为模型，图3-8为观看直播学习活动图。

观看直播学习活动图中，操作对象包括学生用户和教师用户。整个操作活动的具体流程如下：用户登陆到系统中，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，系统返回该学校内开通的所有课程，用户选择课程，进入课堂学习环节，如果系统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。课堂的直播学习是直播页面从流媒体服务器拉流，直播流由学校邀请的教师通过OBS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。



图3-8 观看直播学习活动图

#### 用户升级与创办学校行为模型

用户升级与创办学校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的功能模块。在此以教师用户创办学校为例描述其行为模型，图3-8为教师用户创建学校活动图。



图3-9 创办学校活动图

教师用户登陆系统后，进入用户信息界面，选择创办学校，系统检查是否拥有建校资格，若没有则弹出支付页面，购买成功后继续选择创办学校，进入学校信息录入界面，用户填入相关信息，提交之后系统会检查信息的有效性，例如是否创办过同名称学校等。检查信息合格，则系统提示创办成功，用户信息页面会显示已创办的学校若检查信息不合格，则系统提示失败，创建失败的学校不会显示在用户信息中。

#### 课程管理行为模型

课程管理为学生用户提供购买课程服务，为教师用户提供开通课程服务，同时为两类用户提供课程的基本管理服务，包括课程信息的查看与更改，以及课程的删除。在此以教师用户开设新课程为例描述其行为模型，图3-8为开通新课程活动图。

教师用户登陆系统后，进入用户信息页面，在已创办的学校内选择开通新课程，此时，系统会检查当前用户是否拥有开通新课程的资格。若没有，则弹出支付页面，花费指定金额的虚拟货币，支付成功后重新选择开通新课程。进入开通课程页面后，输入课程相关信息，例如课程参与费用、课程所属段位、课程开通于截至时间等，输入完成后提交，系统会检查课程信息的有效性和课程在所属学校的唯一性。若检查合格，系统提示创建成功并提示课程的推流地址，返回用户信息页面可以看到新的课程已经进入课程管理表项；若检查失败，系统提示开通失败，返回用户信息页面，失败课程没有进入课程管理表项。



图3-10 开通新课程活动图

## 本章小结

本章研究了围棋教学系统的总体需求，分析了完成围棋教学平台所需要的大概模块划分以及完成围棋教学系统所需要的系统角色。通过需求描述和用例图可知围棋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来组合成基本信息与账户管理、在线课堂学习、用户升级与创办学校、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。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。

# 基于围棋的围棋教学系统的设计

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，系统的大致需求已经十分清楚。本章在此基础之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详细表内字段的物理数据模型。

## 围棋教学系统的概要设计

### 围棋教学系统的软件体系结构

围棋教学系统为B/S模式，学生用户只需安装浏览器，即可通过Internet访问本系统；教师用户开设课程的推流环节需要通过OBS等软件将PC截屏的视频流或者通过ffmpeg等将摄像头采集的视频流信息推送到系统指定的RTMP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 系统软件体结构设计

#### 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做以及对课堂互动时聊天信息发布和订阅的支持。对于数据库的基本操作包括对用户信息、开设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支持的的相关频道完成。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。

#### 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功能。业务逻辑主要包括基本信息与账户管理，在线学习，用户升级与创办学校，课堂管理4大功能模块的业务导向。各模块的逻辑服务具体如下：

基本信息与账户管理：用户的注册、登陆、信息更新以及用户钱包的充值和提现；

在线学习：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；

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；

课程管理：学生用户和教师用户对于观看课程的添加、查看、删除，教师用户的开课权限购买和发布新课程，以及对所开课程的信息更新、信息查看、和课堂删除。

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。

#### 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。

### 围棋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账户管理，在线学习，用户升级与创办学校，课堂管理四大模块，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，围棋教学系统通过用户的基本信息和账户管理模块维护用户的信息，包括新用户的注册，用户的信息修改，以及用户虚拟账户的充值和提现；通过在线学习，用户可以选择段位，观看免费视频；选择段位和学校，进入直播课堂，在课堂中，用户可以体验到与教师的实时交流，下载课件，上传等服务；系统通过用户升级与创办学校模块，学生用户可升级为教师用户，有围棋教学经验的教师可开设学校，这两种服务均需支付费用；学生和老师对于自己加入的课堂可以通过课堂管理模块进行维护，其中，教师用户对于自己开设的课堂拥有免费加入权限，无需购买，对于自己开设的课堂，也可进一步更新开课时间等内容。

## 围棋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。

### 基本信息与账户管理

#### 静态结构详细设计

基本信息与账户管理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账户。用户身份主要由学生和教师组成，分别维护着不同的基础信息；虚拟账户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。改模块的类图如图4-3所示。

图4-3 基本信息与账户管理类图

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属性维护了们的虚拟账户的信息。系统通过RegisterView类来接受用户的注册，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，verfy\_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，例如用户可修改邮箱等；账户管理的功能通过WalletView完成，系统通过WalletView.updatemoney(num)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。

#### 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是用户注册时的类间交互。



图4-4 用户注册时序图

用户注册时，会触发注册页面的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，urlpattrn中规则为：url(r’^LearingGo/register.html’,RigsterView.RegisterView.as\_view()),RegisterView在post函数内通过addStudentinfo()填充Ustudent类的对象，通过save()方法存入数据库中并返回存储是否成功的状态，最终register.ajax文件内函数回做出判断，存储成功跳转至登陆页面，存储失败会有“存储失败”的提示，并留在当前网页。

### 在线学习

#### 静态结构详细设计

在线学习是围棋教学系统中最为重要的一部分，围棋教学是围棋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4-5所示。



图4-5 在线学习类图

在线学习由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；参与直播课堂需要用户购买参与权限，由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。参与课堂的围棋学习同样需要选择段位，由IndexView返回合适段位的CampusesView对象列表，进入感兴趣的学校之后选择学校内开设的围棋教学课堂。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。

#### 类间交互详细设计

参与直播课堂时，用户具体的上传作业需要前后端的相互配合完成，具体的类间交互如图4-6所示。



图4-6 上传作业时序图

用户上传作业时，会触发上传页面的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，urlpattrn中规则为：url(r’^LearingGo/hlsroom/(?P<school\_name>)/(?P<class\_name>)/(?P<file\_name>)/$’,HomeworkUpView.HomeworkUpView.as\_view()),HomeworkUpView在post函数内通过addHomework()填充Homework类的对象，通过save()方法存入数据库中并返回存储文件的父路径的状态，HomeworkUpView会查询该目录下的所有文件，返回到前端，homework\_upload()函数会以超链接的形式返回该目录下的所有文件，即所有的上传课件和作业。

### 用户升级与创办学校

#### 静态结构详细设计

用户升级与创办学校使拥有围棋教学资格的用户可以在系统内开建围棋学校，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，提交申请资料并通过后转为教师用户；创办学校只针对教师用户。用户升级与创办学校的类图如图4-7所示。

由图4-7可知，BuyUpgradeView负责学生用户购买升级权限，其中Ustudent的can\_upgrade属性负责用户是否购买了升级权限。拥有升级权限后，通过UpgradeView完成用户升级，其中函数remove\_Stu\_add\_Tea()将删除数据库中存储的学生用户，并提取基本信息后添加教师用户。教师用户的can\_createschool负责用户是否拥有建校权限，通过BuyCreateCampusView支付相关费用后，拥有创建学校的权利，之后可以通过AddCampusView完成campus类的实例化和数据库存储，完成建校。



图4-7 用户升级与创办学校类图

#### 类间交互详细设计

学生用户拥有升级权限之前需要完成向系统支付一定量的虚拟货币，在此以用户购买升级权限为例说明用户升级模块的类间交互，图4-8为用户购买升级权限的时序图。



图4-8 用户购买升级权限时序图

用户点击升级时，会触发页面的buy-upgrade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，urlpattrn中规则为：url(r’^LearingGo/buyupgrade.html’,BuyUpgradeView.BuyUpgradeView.as\_view()),BuyUpgradeView在post函数内通过update\_student()将登陆系统的学生用户对象的can\_upgrade属性设置为非0并完成虚拟货币的扣除，通过update()方法修改数据库中的对应数据并返回更新是否成功的状态，最终buyupgrade.ajax文件内函数回做出判断，更新成功提示“购买成功”，更新失败会有“购买失败”的提示，并留在当前网页。

### 课堂管理

#### 静态结构详细设计

课堂管理主要由“用户参与的课堂管理”和“教师维护的课堂管理”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。用户参与的课堂管理包括课堂的购买加入、信息查看和课堂删除，教师维护的课堂管理包括课堂开设权限的购买，课堂的开设，课堂信息的查看和更新以及课堂的删除。如图4-9为课堂管理的类图。

图4-9 课程管理类图

用户购买课程通过BuyClassView类处理，BuyClassView.add\_Class()函数完成用户虚拟账户的扣费和向syudent\_class或者teacher\_clss添加数据的功能，用户删除课程通过DelClassView完成，DelClassView完成student\_class或teacher\_class内对应课程数据项的删除。教师用户通过BuyCreateClassView完成开课权限的购买，由BuyCreateClassView.update\_campus()完成对campus对象内can\_createschool属性的更新，以及uteacher对象虚拟账户umoney的扣费更新；教师用户购买成功之后拥有开设新课程的权限，通过AddClassView处理教师用户提交的建校信息，通过AddClassView.add\_campus\_class完成classroom新数据的添加。教师用户删除课程关系到classroom表，teacher\_class表和student\_class表的更新，分别由TeaDelClassViewf对象内delete\_classroom()函数，delete\_teacher\_class()和delete\_student\_class()函数负责完成，实现了与此开设课程所有相关课程表的删除。

#### 类间交互详细设计

参与用户删除课程需要用户点击相关课程的删除按钮，经过前段的请求，后端做出处理后完成课程的持久化删除，下次登陆时不再显示。用户删除参与课程的时序图如图4-10所示。



图4-10 用户删除参与课程时序图

用户点击删除课程时，会触发注册页面的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，urlpattrn中规则为：url(r’^LearingGo/delclass/{classname}’,DelClassView.DelClassView .as\_view()),

DelClassView在post函数内通过delete\_class()完成student\_class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，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，删除失败会有“删除失败”的提示，并留在当前网页不变化。

## 围棋教学系统的数据库设计

上一节对围棋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。

### 围棋教学系统的概念数据模型

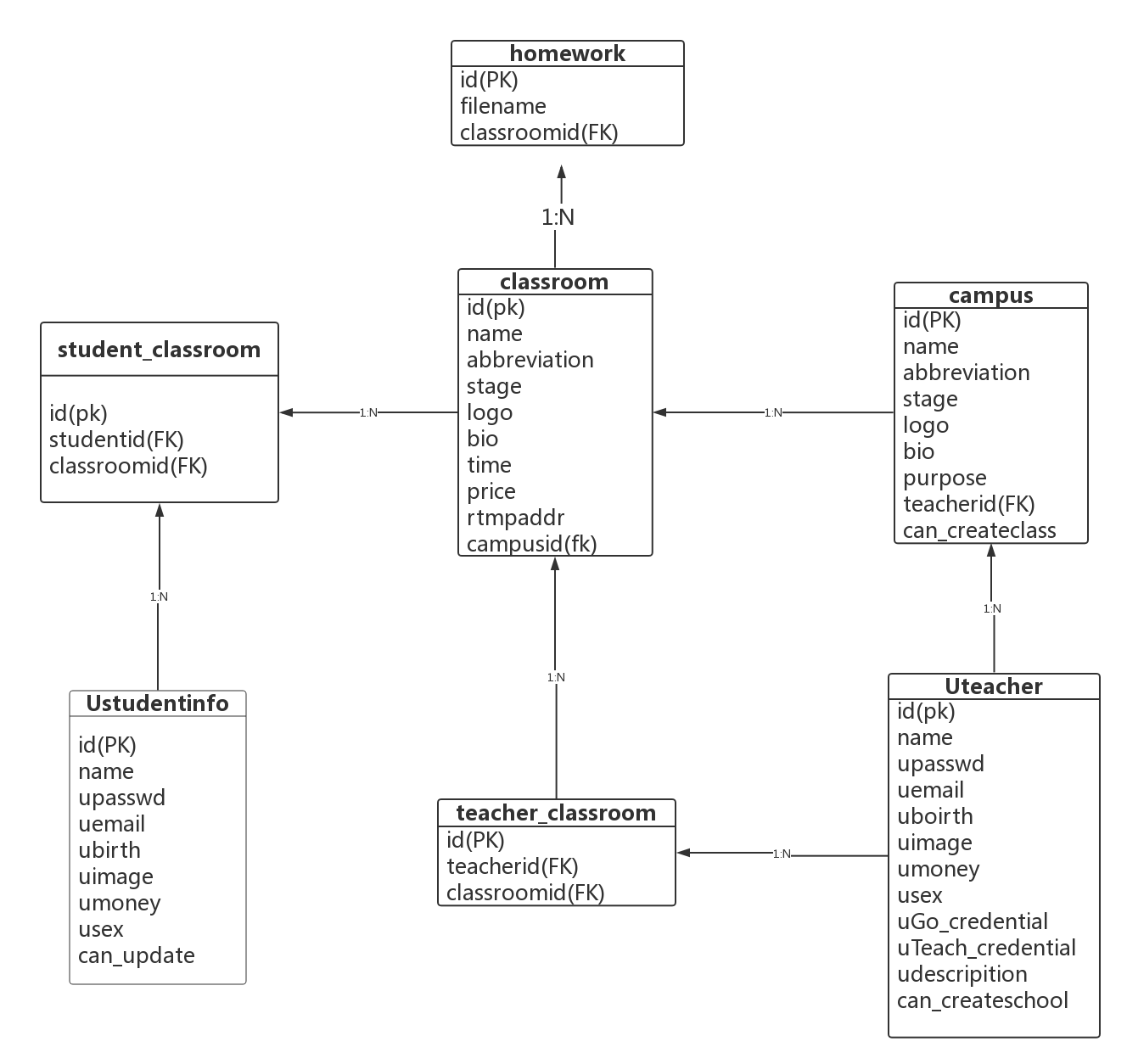
本系统中的数据主要是学生用户，教师用户，校园信息以及课程信息的数据，其中还包括学生参与的课堂信息，教师参与的课堂信息。系统维护这些数据需要通过规范的设计来降低冗余性。由于这些数据基本稳定以及表结构的合理性，系统对这些数据的维护也只需要一些常规性的操作。系统的数据库E-R图如图4-11所示。

图4-11 系统E-R图

### 围棋教学系统的物理数据模型

上一小节对数据库的实体E-R图进行了详细的设计，这一小节将对具体的数据库表进行详细设计，如下所示为数据库表详细设计说明：

#### 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(30) |  | 学生昵称 |
| UWMAIL | Varchar(30) |  | 邮箱地址 |
| UBIRTH | Date |  | 生日 |
| UIMAGE | Varchar(30) |  | 用户头像地址 |
| USEX | Varchar(30) |  | 性别 |
| CAN\_UPDATE | Int |  | 是否有升级权限 |

#### 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(30) |  | 教师昵称 |
| UWMAIL | Varchar(30) |  | 邮箱地址 |
| UBIRTH | Date |  | 生日 |
| UIMAGE | Varchar(30) |  | 用户头像地址 |
| USEX | Varchar(30) |  | 性别 |
| CAN\_CREATESCHOOL | Int |  | 是否有建校资格 |
| UGO\_CREDENTIAL | Varchar(30) |  | 围棋段位水平证明图像地址 |
| UTEACH\_CREDENTIAL | Varchar(30) |  | 教学水平证明图像地址 |
| UDESCRIPTION | Text |  | 教学描述 |

#### 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(30) |  | 学校名称 |
| ABBREVIATION | Varchar(30) |  | 学校英文简称 |
| STAGE | Int |  | 所属段位 |
| LOGO | Varchar(30) |  | 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\_CREATECLASS | Varchar(30) |  | 是否有开设课堂资格 |

#### 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(30) |  | 课堂名称 |
| ABBREVIATION | Varchar(30) |  | 课堂英文简称 |
| STAGE | Int |  | 所属段位 |
| LOGO | Varchar(30) |  | 课堂LOGO地址 |
| BIO | Text |  | 课堂描述 |
| TIME | Text |  | 开课时间说明 |
| PRICE | Int |  | 价格 |
| RTMPADDR | Varchar(30) |  | 课堂推流地址 |
| CAMPUSID | Int | 外键 | 所属学校ID |

#### 课件-作业信息表

表4-5 课件-作业信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 字段类型 | 主键/外键 | 含义 |
| ID | Int | 主键 | 课件或作业标识标识 |
| NAME | Varchar(30) |  | 文件名称 |
| CLASSROOMID | Int | 外键 | 所属课程ID |

#### 视频信息表

表4-6 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(30) |  | 视频文件名称 |
| STAGE | Int |  | 所属段位 |
| LOGO | Varchar(30) |  | 视频图像地址 |
| ADDR | Varchar(30) |  | 视频文件地址 |

#### 学生-参与课堂信息表

表4-7 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 |

#### 教师-参与课堂信息表

表4-8 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 |

## 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。

# 围棋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展示系统的功能。

## 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 |

## 围棋教学系统的实现

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教学系统的实现。系统的实现主要通过对围棋教学系统后端服务器的基本配置和各功能模块的代码实现类来具体说明。

### 后端服务器配置

#### Nginx配置

由于围棋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，接受用户的链接请求，然后将不同的链接服务发送到Nginx+RTMP配置的流媒体服务器，或者uWSGI业务逻辑服务器。Nginx的配置如下

rtmp {

#rtmp 流媒体服务器

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 /tmp/live;

}

#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;

}

#直接播放的本地视频地址

application play {

play /home/loading\_21th/LearningGo/videos;

}

}

}

http {

include mime.types;

default\_type application/octet-stream;

#hls server配置

server {

#直播流监听端口

listen 8081;

location /hls {

types {

application/vnd.apple.mpegurl m3u8;

}

root /tmp;

add\_header Cache-Control no-cache;

}

#hls直播地址

location /live {

alias /home/loading\_21th/LearningGo/golearnApp;

index play.html;

}

#推流拉流状态展示

location /stat {

rtmp\_stat all;

rtmp\_stat\_stylesheet stat.xsl;

}

location /stat.xsl {

root /home/loading\_21th/www;

}

#流媒体控制

location /control {

rtmp\_control all;

}

location / {

alias /home/loading\_21th/www/;

index play.html;

}

}

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

}

location /static {

alias /home/loading\_21th/LearningGo/static/;

}

location /favicon.ico {

root /home/loading\_21th/LearningGo/static/favicon.ico;

}

}

}

#### uWSGI服务器配置

uWsgi服务器负责围棋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

#### 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。本系统的路由配置如下：

url(r’^LearingGo/register.html’,RigsterView.RegisterView.as\_view())

url(r’^LearingGo/login.html’,LoginView.LoginView.as\_view())

url(r’^LearingGo/index.html’,IndexView.IndexView.as\_view())

url(r’^LearingGo/index/videos/(?P<stage>\d+)’,FindVideosView.FindVideosView.as\_view())

url(r’^LearingGo/index/campuses/(?P<stage>\d+)’,FindCampusesView.FindCampusesView.as\_view())

url(r’^LearingGo/campus/(?P<campus\_name>.\*)’,CampusView.CampusView.as\_view())

url(r’^LearingGo/campus/buyclass/(?P<campus\_name>.\*)/(?P<class\_name>)/’,BuyClassView.BuyClassView.as\_view())

url(r’^LearingGo/hlsroom/(?P<campus\_name>.\*)/(?P<class\_name>)/’,HlsRoomView.HlsRoomView.as\_view())

url(r’^LearingGo/hlsroom/homework\_upload/(?P<campus\_name>.\*)/(?P<class\_name>)/’,HomeworkUpView.HomeworkUpView.as\_view())

url(r’^LearingGo/hlsroom/courseware\_download/(?P<campus\_name>.\*)/(?P<class\_name>)/’,CoursewareDownView.CoursewareDownView.as\_view())

url(r’^LearingGo/userinfo/student’,StudentView.StudentView.as\_view())

url(r’^LearingGo/userinfo/wallet’,WalletView.WalletView.as\_view())

url(r’^LearingGo/userinfo/updateinfo’,UpdateinfoView.UpdateinfoView.as\_view())

url(r’^LearingGo/userinfo/buyupgrade’,BuyUpgradeView.BuyUpgradeView.as\_view())

url(r’^LearingGo/userinfo/upgradeinfo’,UpgradeinfoView.UpgradeinfoView.as\_view())

url(r’^LearingGo/userinfo/delclass/(?P<class\_room>)’,DelClassView.DelClassView.as\_view())

url(r’^LearingGo/userinfo/teacher’,TeacherView.TeacherView.as\_view())

url(r’^LearingGo/userinfo/buycreatecampus’,BuyCreateCampusView.BuyCreateCampusView.as\_view())

url(r’^LearingGo/userinfo/addcampus’,AddCampusView.AddCampusView.as\_view())

url(r’^LearingGo/userinfo/updateclass/(?P<campus\_name>.\*)/(?P<class\_name>)/’,UpdateClassView.UpdateClassView.as\_view())

url(r’^LearingGo/userinfo/teacherdelClass/(?P<campus\_name>.\*)/(?P<class\_name>)/’,TeacherDelClassView.TeacherDelClassView.as\_view())

url(r’^LearingGo/userinfo/addclass/(?P<campus\_name>.\*)/(?P<class\_name>)/’,AddClassView.AddClassView.as\_view())

### 功能模块的实现

#### 基本信息与账户管理模块

基本信息与账户管理模块中，系统主要维护学生用户和教师用户的基本信息和虚拟账户的充值与提现。其中，用户信息管理模块的用户注册中，注册成功的用户默认为学生用户，通过用户升级可以成为教师用户，教师用户的基本信息和账户也在此模块的功能之中。用户注册的核心代码如下：

class RegisterView(View):

'''添加用户，注册默认是学生'''

def add\_Studentinfo(name,upasswd,uemail,ubirth,usex,uimage,umoney=0,

can\_upgrade=False):

student = BaseTable.Ustudentinfo(name=uname,upasswd=passwd,uemail=email,ubirth=birth,

usex=sex,uimage=file\_obj)

student.save()

@csrf\_exempt

def post(self,request):

if request.FILES:

'''用户头像'''

file\_obj = request.FILES.getlist('filename')[0]

uname = request.POST.get('name')

passwd = request.POST.get('passwd')

email = request.POST.get('email')

birth = request.POST.get('birth')

sex = request.POST.get('sex')

add\_Studentinfo(name,upasswd,uemail,ubirth,usex,uimage)

…

return response

通过代码可知，用户注册时，通过post提交注册信息，系统取得各项信息后实例化Ustudentinfo对象，通过Ustudentinfo的save()方法存储到数据库中在之后登陆或者查看用户信息的过程，可以取得该数据进行匹配或查看用户信息。

#### 在线学习模块

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，由于各围棋课堂属于学校的运营管理范围之内。因此，直播流的推流拉流地址末尾以学校名和课堂名作为划分，例如：RTMP://192.168.15.20:1935:/xjtu/soft/。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 HomeworkUpView(View):

def add\_Homework(school\_name,class\_name,fileobj):

'''homework对象的存储路径根据学校名和课程名动态变化'''

homeworkfile= Homework(schoolname=school\_name,classname=class\_name,homework=file\_obj)

homeworkfile.save()

@csrf\_exempt

def post(self,request,school\_name,class\_name):

'''文件上传目录'''

upload\_fpath = os.path.dirname(os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_))))

upload\_path = os.path.join(upload\_fpath,'uploadfile',school\_name,class\_name)

if request.FILES:

'''获取文件对象'''

file\_obj = request.FILES.getlist('filename')[0]

add\_Homework(school\_name,class\_name,file\_obj):

'''返回存储目录下所有文件名'''

result = {'Courseware\_name':os.listdir(upload\_path)}

…

return response

由代码可知，用户上传作业时，系统通过post方法接受post传送的数据，通过该数据实例化Homework对象进行存储。由前一节的url分发路由配置可知，向/LearingGo/hlsroom/homework\_upload/(?P<campus\_name>.\*)/(?P<class\_name>)提交post请求时，Django的URLConf会将匹配的政策表达式中的campus\_name和class\_name作为post的参数进行传递，由此实现了文件上传的隔离性。

#### 用户升级与创建学校模块

用户升级是由学生用户转为教师用户的唯一办法，拥有教师身份，方可创办学校，建立课堂。用户升级，创办学校过程中，系统采取收取虚拟货币，验证提交信息的步骤。用户升级过程中，需要判断用户是否缴纳费用，付费之后才拥有升级权限。升级成功后，需要将学生用户转为教师用户。教师用户创建学校同样需要缴纳一定的虚拟货币才有建校资格。 在此，以用户升级过程为例展示代码。

class BuyUpgradeView(View):

def get(self,request):

name = request.session.get('name')

student = BaseTable.Ustudentinfo.objects.get(name=name)

result = {'status':'fail'}

'''判断是否有权限升级'''

if student.can\_upgrade :

result['status'] = 'success'

response = JsonResponse(result, safe=False)

response["Access-Control-Allow-Origin"] = "\*"

response["Access-Control-Allow-Methods"] = "POST, GET, OPTIONS"

response["Access-Control-Max-Age"] = "1000"

response["Access-Control-Allow-Headers"] = "\*"

return response

@csrf\_exempt

def post(self,request):

student = BaseTable.Ustudentinfo.objects.get(name=request.session['name'])

nowmoney = student.umoney

money\_update\_sum = request.POST.get('money\_update\_sum')

stat = "操作失败，请重试"

if (request.POST.get('is\_add')== "yes"):

nowmoney = nowmoney + int(money\_update\_sum)

else:

nowmoney = nowmoney - int(money\_update\_sum)

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:

'''完成支付，设置可以升级的属性'''

stat = "success"

student.umoney = nowmoney

if (request.POST.get('is\_add')== "pay\_to\_sys"):

student.can\_upgrade = True

student.save()

…

return response

class UpgradeinfoView(View):

def delete\_student\_by\_name(name):

student = BaseTable.Ustudentinfo.objects.get(name=name)

student.delete()

def add\_Teacher\_from\_studentname(name,Go,teach,desc):

student = BaseTable.Ustudentinfo.objects.get(name=name)

teacher = BaseTable.Uteacherinfo(

name=student.name,upasswd=student.upasswd,

uemail=student.uemail,ubirth=student.ubirth,

uimage=student.uimage,umoney=student.umoney,

usex=student.usex,uGo\_credential=Go,

uTeach\_credential=teach,

udescripition=desc,can\_createschool=False)

teacher.save()

@csrf\_exempt

def post(self,request):

name = request.session.get('name')

Go = request.FILES.getlist('uGo\_credential')[0]

teach = request.FILES.getlist('teach\_credential')[0]

desc = request.POST.get("descripition")

'''升级学生用户为教师用户，添加教师数据'''

add\_Teacher\_from\_studentname(name,Go,teach,desc)

'''删除学生数据'''

delete\_student\_by\_name(name)

…

return response

由上述代码可知，BuyUpgradeView负责判断用户是否拥有升级权限，并可通过post方法进行购买，虚拟货币流向系统账户，并更新了学生用户的can\_upgrade属性。用户升级信息的提交和审核通过UpgradeinfoView的post方法处理，拥有权限的学生可提交申请信息，系统处理后，提取学生信息并加入到教师用户的数据表中，删除学生数据，完成用户升级。

#### 课程管理模块

课堂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，课堂管理模块还包括教书用户对自己所开设课堂的增删改查的权限，在此以教师用户开设新课为例说明课堂管理模块的实现过程。

class AddClassView(View):

def update\_can\_createclass(campus,stat):

campus.can\_createclass = stat

campus.save()

def add\_class(teacher,name,abbr,stage,logo,bio,time,price,campus):

rtmp = "rtmp:192.168.15.12:1935/"+campus.abbreviation+"/"+abbreviation

class = Classroom(name=name,abbreviation=abbr,stage=stage,logo=logo,bio=bio,time=time,price=price,rtmpaddr=rtmp,campus=campus)

class.save()

add\_Teachers\_Classer(teacher,class)

return rtmp

'''教师所开课程拥有免费观看权限'''

def add\_Teachers\_Classes(teacher,class):

tea\_class = Teachers\_Classes(teacher=teacher,classroom=class);

tea\_class.save()

@csrf\_exempt

def post(self,request,school\_name):

name = request.session.get('name')

teacher = BaseTable.UTeacherinfo.objects.get(name=name)

campus = BaseTable.Campus.objects.get(name=school\_name)

'''添加课程信息'''

name = request.POST.get("name")

abbr = request.POST.get("abbr")

stage = request.POST.get("stage")

logo = request.FILES.getlist('logo')[0]

bio = request.POST.get("bio")

time = request.POST.get("time")

price = request.POST.get("price")

rtmp = add\_class(teacher,name,abbr,stage,logo,bio,time,price,campus)

'''开课次数用完，更新开课权限'''

update\_can\_createclass(campus,False)

result = {'rtmpaddr':rtmp}

…

return response

由上述代码可知，教师用户开设新课同样需要先后买开课权限，拥有权限之后方可提交课堂信息，由于课堂属于学校的经营管理范围之内，因此增加课堂需要学校的信息。由前一节的url分发路由配置可知，向/LearingGo/hlsroom/homework\_upload/

(?P<campus\_name>.\*)提交post请求时，Django的URLConf会将匹配的政策表达式中的campus\_name作为post的参数进行传递，由此实现了传送学校的信息课堂开设时，需要有独立的rtmp推流地址，系统根据所开课堂的学校简称和该课堂的简称分配rtmp地址，并在课堂开设成功后返送给教师用户。教师在所设时间内，将直播流推送到指定地址即可实现实时的课堂直播。

## 围棋教学系统的测试

### 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OBS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,

OBS Studio 20.1.3

1. 服务端

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。

### 功能性测试

在上一节客户端和服务器端的测试环境介绍完成之后，本节将说明对围棋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](http://baike.baidu.com/view/17674.htm)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  账户管理 | 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 |
| 在线学习 | 选择段位观看视频 | 各段位视频可以观看 | 各段位视频可以观看 | 通过 |
| 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 |
| 下载课件 | 下载课件成功 | 下载课件成功 | 通过 |
| 用户升级  与  创建学校 | 验证升级权限 | 是否有升级权限 | 是否有升级权限 | 通过 |
| 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 |
| 建校信息提交 | 建校成功 | 建校成功 | 通过 |
| 课程管理 | 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 |
| 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 |
| 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 |

## 本章小结

本章主要介绍说明了围棋教学系统的开发测试情况。通过对围棋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，详细介绍了围棋教学系统的实现过程和测试过程，完整的展现了围棋教学系统的开发、运行、测试的情况。

# 结论与展望

## 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，同时作为一项长盛不衰的体现人类智慧的竞技类运动，近年来越来越受到大众的关注，很多人都在业余时间去专门的学习围棋，也有很多人将自己的孩子送到围棋培训班内学习。但是围棋培训市场的教学质量，教学的规范性以及教育资源的局限性使得围棋的发展极为不平衡，甚至对于有浓厚兴趣却得不到优秀教育资源的围棋学子也很不公平。同时，伴随近年来国内互联网的发展和直播行业的崛起，建设一套针对围棋教学的直播系统来平衡围棋教学资源，解决教育设施的地域局限性，加强围棋教学的规范性十分有必要。同时，围棋教学系统在得以线上展开，也将带来一个极具经济规模的市场。考虑到上面的要求，主要工作内容包含以下几点：

1）具体分析了围棋教学系统的需求。结合围棋教学的现实，围棋教学系统应该将围棋教学做成面向多种段位选手，提供从入门到高级选手学习课程的立体化教育网站。围棋教学系统为围棋学生用户提供服务，同时，教师用户也可以在系统内较为方便的开展围棋教学。系统维护两类主要用户的基本信息、为用户参与的课堂提供管理功能，教师用户维护自己开设的学校和课堂，且系统提供了免费的视频观看和需付费的课堂教学。

2）系统在需求分析的基础上完成了对系统的详细设计。通过较为宏观的系统软件体系结构和系统的功能模块划分，清楚了系统的内部组织和各模块面向用户提供了什么具体的服务。在各模块的详细设计阶段，通过类图和时序图分析了功能模块内部的结构以及用户接受服务需要通过的类间信息通路。数据的设计保障了系统基本数据的高校操作，避免了数据冗余，也是为系统业务逻辑层的服务提供了底层的数据支持，支撑了系统功能的展开。

3）系统在详细设计的基础上完成了系统实现与系统测试。通过对系统的开发环境、开发服务器配置以及系统内四个模块代码实现的说明，清晰展示了系统的搭建开发情况，展示了系统的运行过程。测试环节说明了系统的测试环境，以及各测试用例的通过情况，说明了系统功能的完整性。

## 展望

当今社会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学习方式的分析，结合当下互联网技术的发展，建设了一套具有可行性的线上围棋教学系统，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技术方面的原因仍然非常有限，因此提出的方法也具有局限性。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，通过本系统的课件学习无法达到要求，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可惜啊的围棋实践学习。

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

# 参考文献

# 附 录

# 攻读学位期间取得的研究成果

学位论文独创性声明（1）

学位论文独创性声明（2）

学位论文知识产权权属声明