ZUCC教师-课程评价系统

**软件需求说明书**

**小组编号：4**

**组长：童鑫聪**

**组员：项伟铭、罗丹妮**

目录

[1引言 4](#_Toc22492)

[1.1目的 4](#_Toc11325)

[1.1.1文档编写目的 4](#_Toc19345)

[1.1.2预期读者 4](#_Toc4000)

[1.2范围 4](#_Toc14082)

[1.2.1产品名称 4](#_Toc6428)

[1.2.2软件的应用 4](#_Toc4414)

[1.3定义、简写和缩略语 5](#_Toc6135)

[1.4引用文件 5](#_Toc5916)

[1.5综述 5](#_Toc32177)

[2总体描述 6](#_Toc1045)

[2.1产品描述 6](#_Toc20460)

[2.2产品功能 6](#_Toc26384)

[2.3用户特点 6](#_Toc10904)

[2.3.1用户类别 6](#_Toc6740)

[2.3.2用户调研记录 7](#_Toc23697)

[2.4约束 10](#_Toc24636)

[2.4.1法规政策 10](#_Toc14857)

[2.4.2硬件局限 10](#_Toc29284)

[2.4.3与其他应用的接口 11](#_Toc23420)

[2.4.4并行操作 11](#_Toc32616)

[2.4.5审核功能 11](#_Toc29948)

[2.4.6控制功能 11](#_Toc13161)

[2.4.7高级语言需求 11](#_Toc4370)

[2.4.8信号握手协议 11](#_Toc2552)

[2.4.9可靠性需求 11](#_Toc5009)

[2.4.10应用的关键性 11](#_Toc23575)

[2.4.11安全和保密安全考虑 12](#_Toc19462)

[2.5假设和依赖关系 12](#_Toc29542)

[2.6需求分配 12](#_Toc23726)

[3具体需求 12](#_Toc29639)

[3.1外部接口需求 12](#_Toc8732)

[3.1.1用户界面 12](#_Toc14064)

[3.1.2界面原型图 13](#_Toc26858)

[15](#_Toc20938)

[3.1.2.2界面原型图说明 17](#_Toc27187)

[3.1.3硬件接口 19](#_Toc31118)

[3.1.4软件接口 19](#_Toc3242)

[3.1.5通信接口 20](#_Toc28413)

[3.2功能需求 20](#_Toc6018)

[3.2.1信息流 20](#_Toc2087)

[3.2.2过程描述 25](#_Toc11657)

[3.2.3数据构建规范 31](#_Toc30109)

[3.2.4数据字典 35](#_Toc23074)

[3.3性能需求 38](#_Toc1540)

[3.4设计约束 38](#_Toc1790)

[3.5软件系统属性 38](#_Toc861)

[3.6其他需求 39](#_Toc10444)

# 1引言

## 1.1目的

### 1.1.1文档编写目的

本文档用于指导软件开发者开发社团管理系统软件的过程。通过规范开发人员的开发过程达到保证项目团队完成软件项目的基本要求，提高软件质量的目的。开发者应根据本文档进行软件开发和编制软件开发文档。开发者应根据本文档进行软件开发和编制软件开发文档。本文档供项目经理、设计人员、开发人员参考。

### 1.1.2预期读者

预期读者：用户代表、项目组成员

## 1.2范围

### 1.2.1产品名称

所开发软件产品名称：ZUCC教师-课程评价平台（ZUCC-COTC）

### 1.2.2软件的应用

1. 目的：

* 用于为学生提供反映教师-课程教学质量的平台
* 方便学生选课时获取参考信息
* 教师可通过该平台了解学生的学习感受，改善教学质量

1. 收益：

*TBD*

## 1.3定义、简写和缩略语

#### 1.3.1产品名称

全写：ZUCC教师-课程评价平台

缩略语：评价平台

简写：ZUCC-COTC ( COTC: Comments On Teachers and Course )

#### 1.3.2教师-课程

简写：教师-课程

含义：由某个教师教授的某门课程，如杨枨老师开设的软件工程可用“杨枨-软件工程”表示，也可用“教师-课程”表示这样的一个组合关系

#### 1.3.3方正教务管理系统

全写：方正教务管理系统

缩略语：xk网

## 1.4引用文件

文件：GB/T 9385-2008,计算机软件需求规格说明规范[S].中国:中华人民共和国国 家质量监督检验检疫总局、中国国家标准化管理委员会,2008.

获取来源：国家标准全文公开系统( openstd.samr.gov.cn )

## 1.5综述

SRS组织方式：按功能层次组织

章节3.1将描述软件系统所有输入和输出

章节3.2将定义软件在接收和处理输入以及处理和产生输出中必须发生的基本动作

章节3.3规定软件或人与软件互作用的整体静态的和动态的数量化需求。

章节3.4规定可能有其他标准、硬件局限等引发的设计约束

章节3.5规定了所要求的软件属性，以客观地验证属性的实现情况

章节3.6规定了可靠性、可用性、安全保密性、可维护性、可移植性等需求

# 2总体描述

## 2.1产品描述

*TODO*

## 2.2产品功能

**普通用户：**

1．用户登录后可以根据课程名搜索课程，根据教师名搜索教师

2．用户登录后可查看教师的整体得分、简介、任教课程、收到的评价

3．用户可查看某门课程有哪些教师开课，可查看有关该课程的评价

4．用户可查看教师-课程的总体得分、点到率、考核方式、平均给分（学生在该教师该课程的平均绩点）、收到的评价

6．用户可对教师-课程作出评价并可对自己的评价进行管理查看或删除评价

7．用户可对他人作出的评价表示“认同/反对”

8．用户可对含有不当内容的言论进行举报

9. 用户可以修改自己的账号信息，如昵称、账号密码等

10. 所有操作得在用户登录状态下进行

**管理员用户：**

1. 管理员根据系统数据库分配账号密码进入登录后再操作
2. 管理员可增删改查基础信息，如教师、课程、用户信息、用户评价等
3. 管理员可对用户所举报的评论进行审核，如审核确为不当评论对该评论用户做警告处理并删除该评论，若该用户被警告多次则设置封号期限

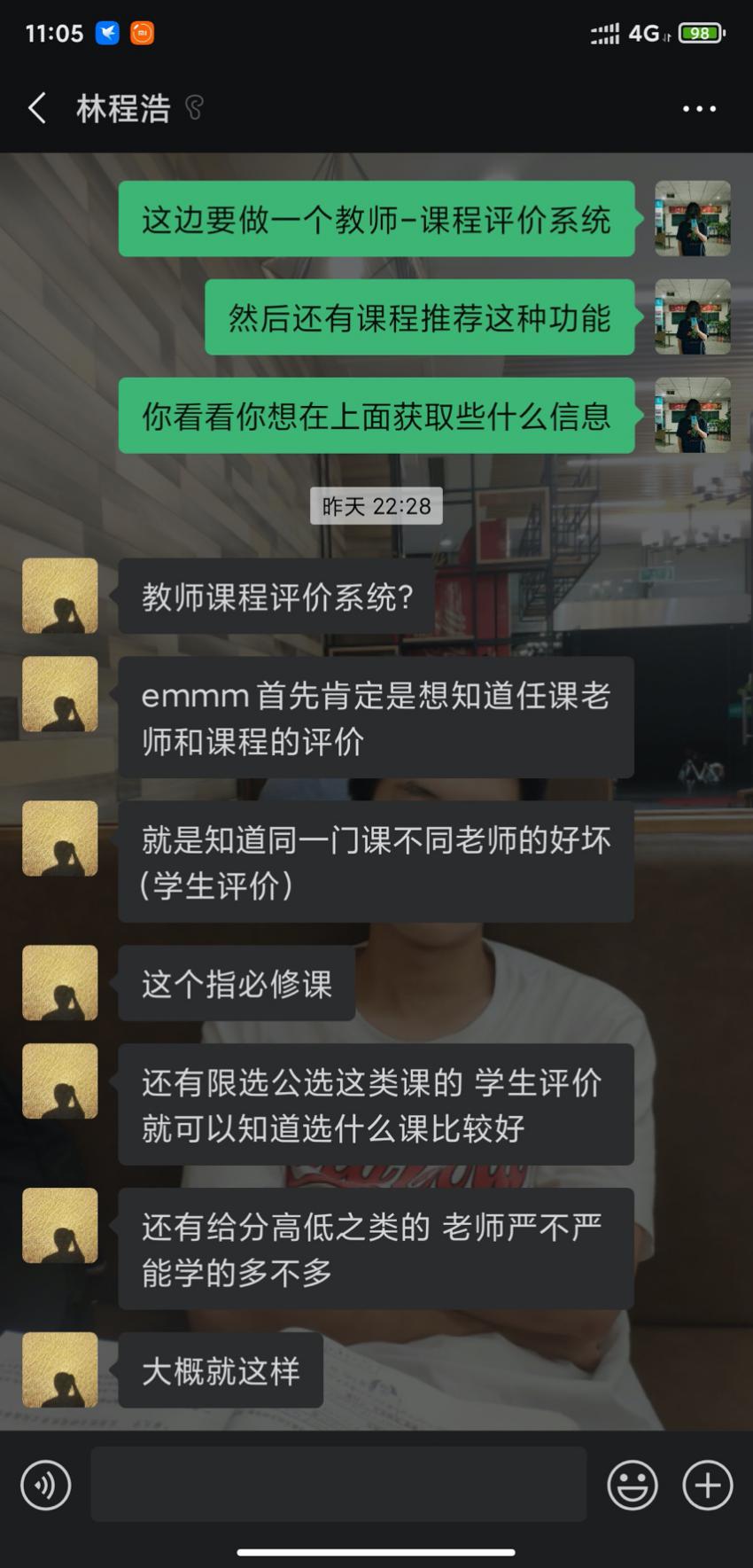
## 2.3用户特点

* 教育程度：本科在读
* 就读学校：浙大城市学院
* 网站访问经验：有过主流网站的使用经验

### 2.3.1用户类别

|  |  |
| --- | --- |
| 用户特征 | 用户代表 |
| 对绩点要求高的学生 | 潘殊焱 软工1802 |
| 抗压能力较差想安稳过的学生 | 张若琪 国贸1803 |
| 课外生活丰富到甚至占用上课时间的学生 | 林某浩 计算1802 |
| 喜欢被严格的老师鞭策想干出一番大事业的学生 | 童峻涛 软工1802 |
| 管理员用户 | 庄博伟 软工1802 |

### 2.3.2用户调研记录



**图2.3.2-1 用户调研记录1**

|  |  |
| --- | --- |
| 采访对象 | 软工1802 潘姝焱 31801328 |
| 采访人员 | 软工1802 罗丹妮 31801326 |
| 采访目的 | 对于搜索界面初步原型设计的看法及建议 |
| 采访时间 | 2020/11/9 21：50~22：10 |
| 采访地点 | 寝室 |
| 采访问题及回答:  1、你觉得该原型设计是否美观？色彩搭配怎么样？  A：颜色不好看 蓝黑色暗色与白色冲突  2、你觉得使用简单吗，界面容易理解吗？  A：挺容易的  3、对该界面你有什么使用上的不满？  A：搜索部分（搜索框）界面展示不够明确，底部栏内容有点空 可以加地址，现有的教师推荐块间隔过大 留白过多  4、你希望增加什么功能（需求）？  A：希望老师具体信息能详细一些便于联系，  5、你有什么好的建议吗（不限制方面）？  A：可以丰富模块的划分，内容完善丰富一些 | |

**图2.3.2-2 用户调研记录2**

|  |  |
| --- | --- |
| 采访对象 | 软工1802 童峻涛 31801341 |
| 采访人员 | 软工1802 罗丹妮 31801326 |
| 采访目的 | 对于教师评分系统初步原型设计的看法及建议 |
| 采访时间 | 2020/11/16 20：40~21：40 |
| 采访地点 | 微信线上 |
| 采访问题及回答:  1、你觉得该原型设计是否美观？色彩搭配怎么样？  A：总体框架完善，设计较合理，封面选图不错，色彩搭配细节需要界面背景颜色再考虑（例如登陆页面黄色提醒字体）  2、你觉得使用简单吗，界面容易理解吗？  A：简单；容易理解  3、对该界面你有什么使用上的不满？  A：有些字体太小看起来困难  4、你希望增加什么功能（需求）？  A：希望能够展示老师的信息，例如办公地点等  5、你有什么好的建议吗（不限制方面）？  A：提升用户与界面的交互；界面的布局可以再考虑优化； | |

**图2.3.2-3 用户调研记录3**

|  |  |
| --- | --- |
| 采访对象 | 国贸1803 张若琪 31805411 |
| 采访人员 | 软工1802 罗丹妮 31801326 |
| 采访目的 | 对于搜索界面初步原型设计的看法及建议 |
| 采访时间 | 2020/11/10 17：45~18：10 |
| 采访地点 | 理四（有录音） |
| 采访问题及回答:  1、你觉得该原型设计是否美观？色彩搭配怎么样？  答：边框栏颜色过于深，整体感觉过于正式，如果调成天蓝色等一些看起来比较随和活泼的颜色会好一些；  背景可以换成网格，减少一些死板；教师介绍间隔过大，留白过多。  2、你觉得使用简单吗，界面容易理解吗？  答：简单，容易理解。  3、对该界面你有什么使用上的不满？  答：平均绩点看不出来（页面缺少一些解释说明），字体太小看起来困难。  4、你希望增加什么功能（需求）？  答：可以显示教师的联系方式（如办公室、电话、邮箱）。  5、你有什么好的建议吗（不限制方面）？  答：界面介绍主题----教师评分系统。 | |
| 补充说明：  对于其它设计界面的初步看法：具体评论页面点老师名字能再次出现老师的详情介绍。 | |

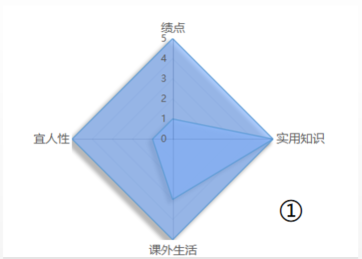
**图2.3.2-4 用户调研记录4**

|  |  |
| --- | --- |
| 采访对象 | 杨枨教授 |
| 采访人员 | 童鑫聪 罗丹妮 项伟铭 |
| 采访目的 | 对于教师——课程评价系统初步原型设计的看法及建议 |
| 采访时间 | 2020/11/16 17：20~17：50 |
| 采访地点 | 理四504 |
| 采访总结：   1. 界面原型不够美观，设计风格及排版有些古老 2. 缺少用户管理员界面 3. 常用评价提供关键词（希望学生进行评价的要素，如上课风格、态度等）可供用户进行选择，提高精准度 4. 动作及详细信息内容（如各个分院名称）补上 5. 对各个页面的一些要素进行解释说明（如平均绩点） 6. 缺少底部栏，课程搜索界面根据课程名称重新制作 | |

**图2.3.2-5 用户调研记录5**

|  |  |
| --- | --- |
| 采访对象 | 软工1802 庄博伟 |
| 采访人员 | 计算1801 童鑫聪 |
| 采访目的 | 对于管理员界面demo的看法和建议 |
| 采访时间 | 2020/11/18 16:14 |
| 采访地点 | 理四 |
| 采访内容：  1、你觉得分院管理这部分的表格中还需要添加哪些字段？  答：没有想法，感觉这么多够了  2、你觉得教师管理这部分的表格中还需要添加哪些字段？  答：字段没什么好加了，但是应该加上搜索功能，不然登记的时候一个老师可能登记过了但是忘了，手动翻的话太慢了  3、你觉得评价管理这部分有哪些欠缺？  答：你这里缺个批量管理，就是同时给多个用户设置禁言或者解禁之类的 | |
| 修改方案：每个管理页面增加搜索功能，增加多选操作批量处理 | |

**图2.3.2-6 用户调研记录6**



**图2.3.2-7 用户调研雷达图1**

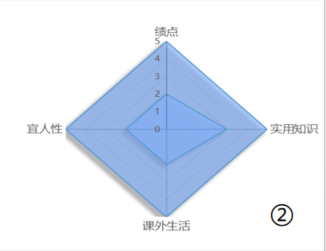


图2.3.2-8 用户调研雷达图2

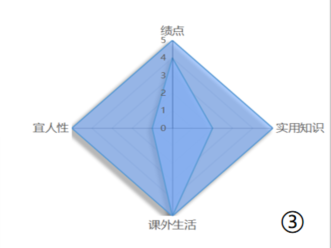


图2.3.2-9 用户调研雷达图3

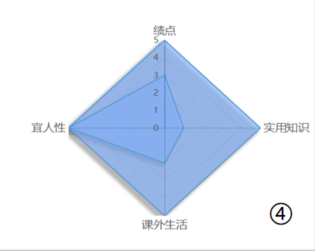


图2.3.2-10 用户调研雷达图4

用户访谈的结果大致如雷达图上显示，用户们对绩点、实用知识、宜人性、课外生活的需求相对比较有差异性，而不同的用户群体积累到一定程度时，其他用户就能在群体中找到适合自己的那一类群体对号入座，找到与他们适配的课程与老师

## 2.4约束

### 2.4.1法规政策

遵循《互联网论坛社区服务管理规定》、《互联网信息服务管理办法》、《计算机信息网络国际联网安全保护管理办法》和《互联网电子公告服务管理规定》等法律法规，不发布违法内容，不泄露用户信息，与用户签订协议明确用户不得发布违法内容，完善审核机制，加强对其用户发布信息的管理。

在获取用户选课信息时需要爬取xk.zucc.edu.cn站点的内容，此时应遵循《数据安全管理办法（征求意见稿）》，尽管该管理办法为征求意见稿，但这是针对网络爬虫的法律法规，有很高的参考价值，遵循该《办法》将使得软件系统在法律层面上更加规范

### 2.4.2硬件局限

*TBD*

### 2.4.3与其他应用的接口

与xk.zucc.edu.cn的接口：获取注册用户在该网站上的选课记录

### 2.4.4并行操作

一个班级算25人，一个专业算2班，一个学院算3个专业，学校有12个学院，最大总用户量在7000左右，实际用户量按2000记，高峰期当日用户量按1000记，所以软件系统要求在高峰时段承受1000用户同时在线，且所有业务动作可用、稳定，页面跳转时长不超过5秒

### 2.4.5审核功能

管理员要能对用户的举报进行处理，审核被举报评论

### 2.4.6控制功能

***TBD***

### 2.4.7高级语言需求

***TBD & TBDLT***

### 2.4.8信号握手协议

***TBD & TBDLT***

### 2.4.9可靠性需求

* 系统应保证7\*24小时不间断运行
* 服务器应在高峰期时实现20分钟内的配置升级
* 平均故障时间间隔应不低于2880h

### 2.4.10应用的关键性

***TBD & TBDLT***

### 2.4.11安全和保密安全考虑

* xk.zucc.edu.cn 采用 HTTP 协议明文传输用户密码，无可避免，有一定的风险泄露用户的xk网密码
* 小程序要求服务器具有SSL证书，HTTPS 能提高服务器与用户间传输信息的安全性

## 2.5假设和依赖关系

* 日活用户的估计 - 影响2.4.4并行操作
* aliyun服务器的重启时间 - 影响2.4.9可靠性需求的离线配置升级时间
* xk网的登陆方式 - 影响（TODO）

## 2.6需求分配

1. 实现登陆、搜索、浏览、留言
2. 加入推荐系统
3. 加入举报
4. 加入整体评分
5. 加入平均给分
6. 加入点名概率

# 3具体需求

## 3.1外部接口需求

### 3.1.1用户界面

有主流网站使用经验的大学生能在3分钟内完成注册登陆，2分钟内搜索到目标教师或课程

采用B/S，用户通过浏览器访问

分辨率不低于1920\*1080

主页：

内容：注册/登陆按钮、搜索框、推荐项、按分院检索菜单

### 3.1.2界面原型图

##### 3.1.2.1界面原型图展示

修改前：



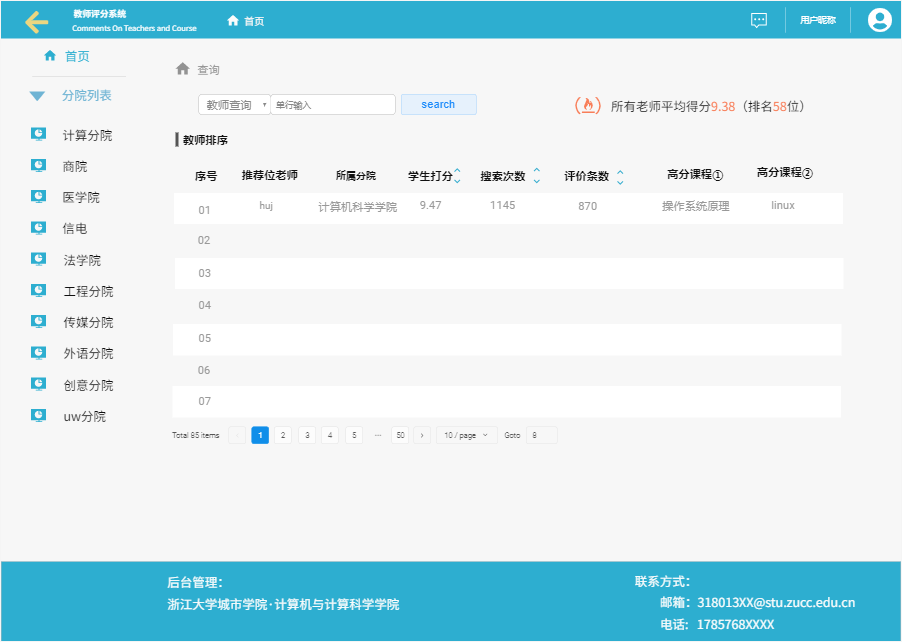
修改后：



**图3.1.2.1-1 登录界面**



**图3.1.2.1-2 搜索主界面**



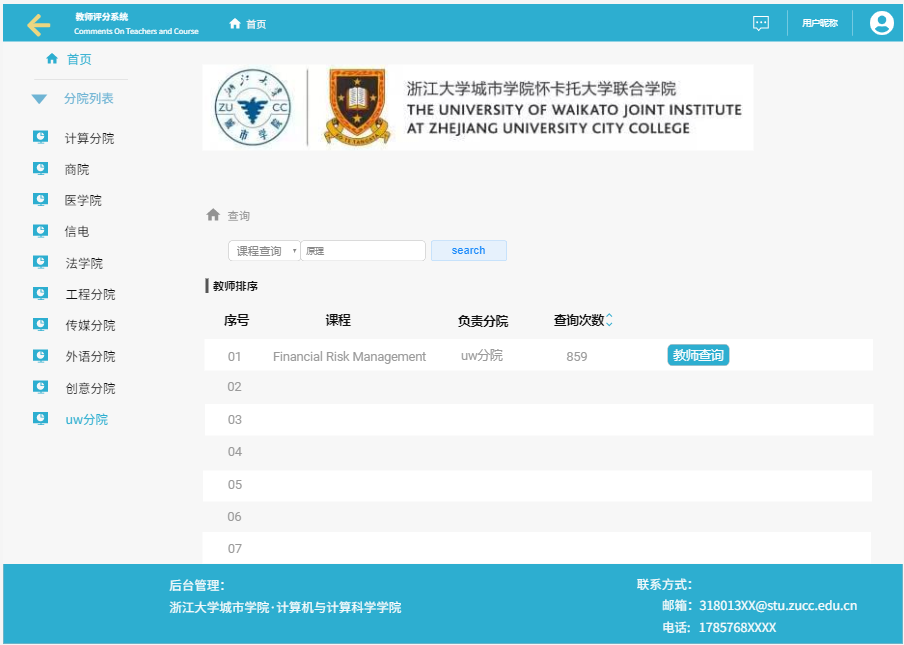
**图3.1.2.1-3 教师搜索主界面**



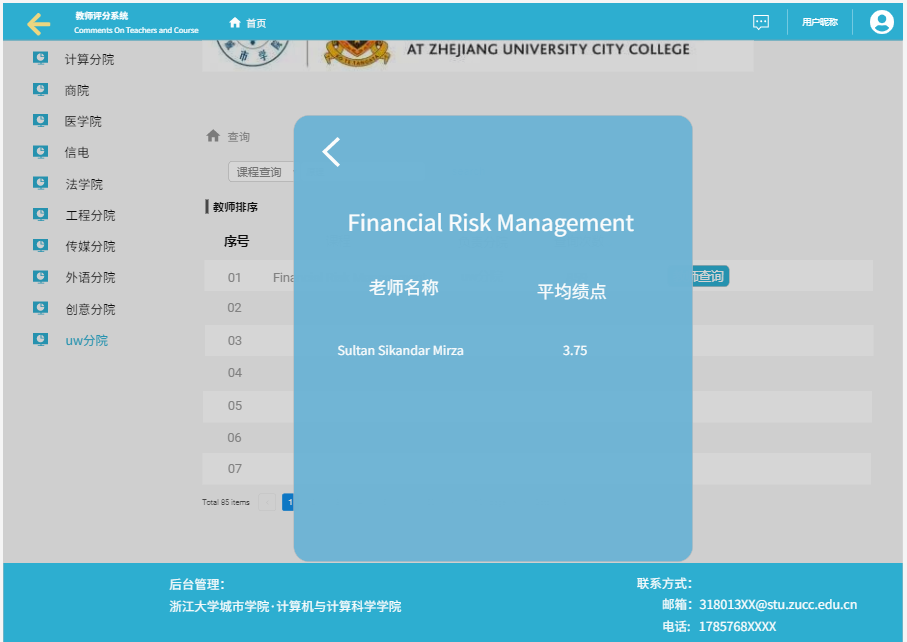
**图3.1.2.1-3 课程搜索主界面**



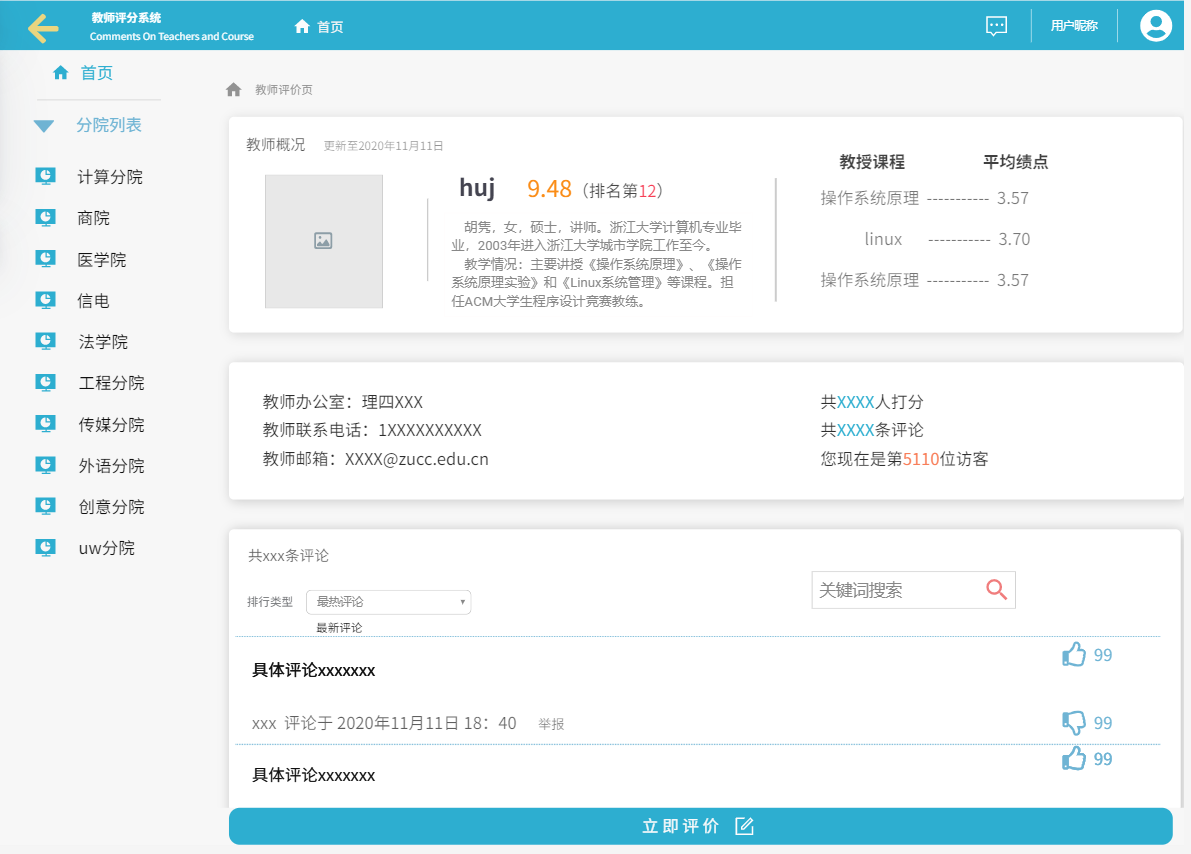
**图3.1.2.1-3 分院教师搜索主界面**



**图3.1.2.1-4 分院——课程搜索主界面**



**图3.1.2.1-5 分院——课程教师搜索主界面**



**图3.1.2.1-6 教师详情介绍界面**



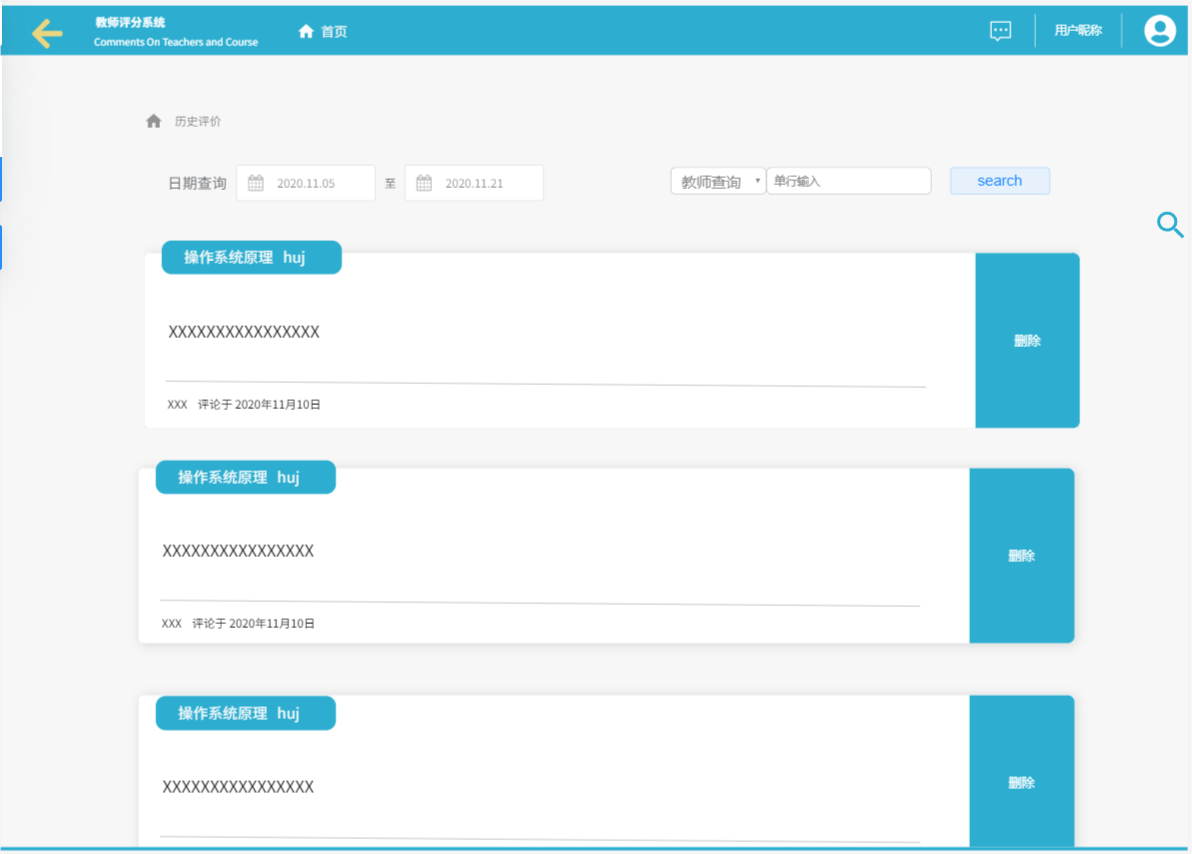
**图3.1.2.1-7 评价界面**

### 

**图3.1.2.1-8 课程——教师选择页**



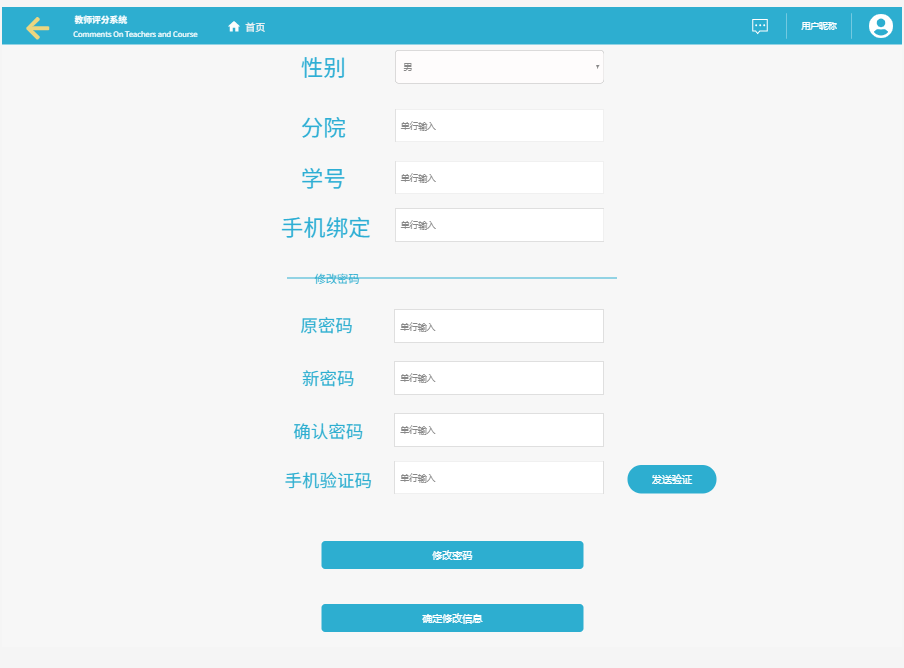
**图3.1.2.1-9 教师——课程详情页**



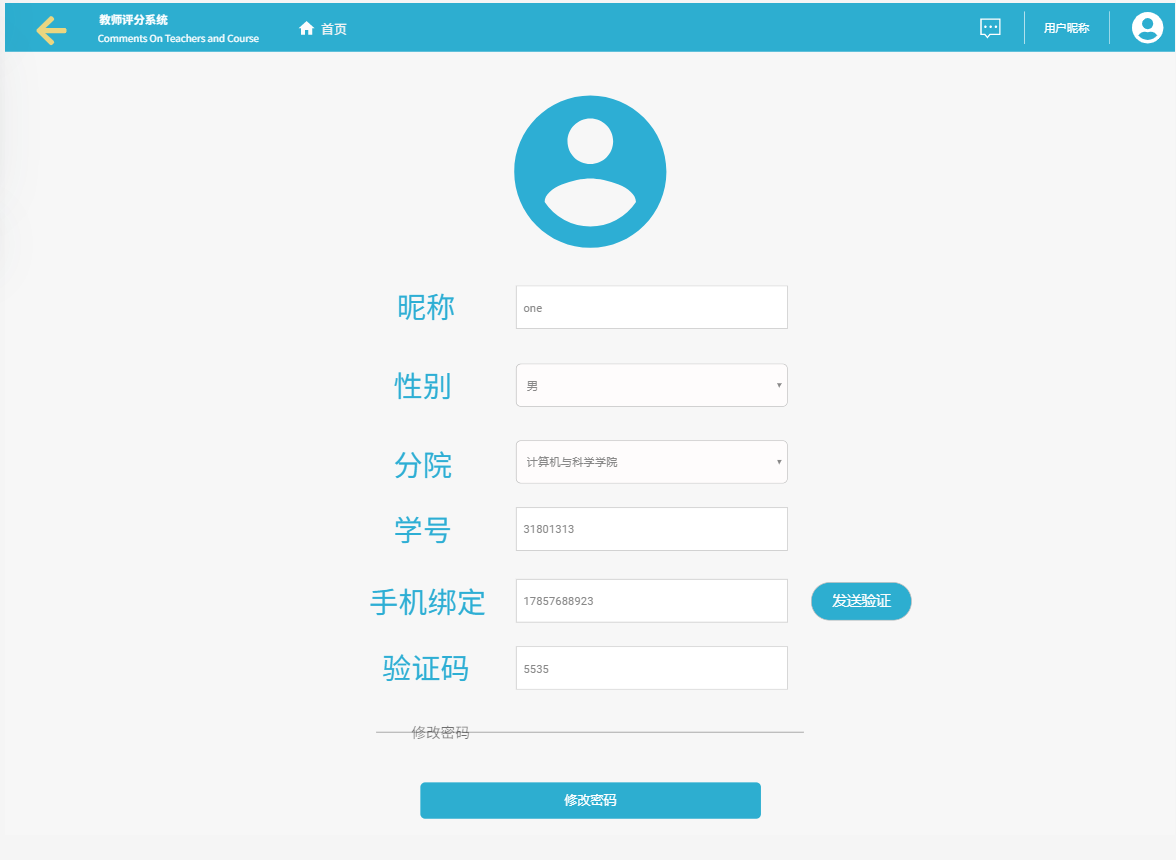
**图3.1.2.1-10 历史记录查询界面**



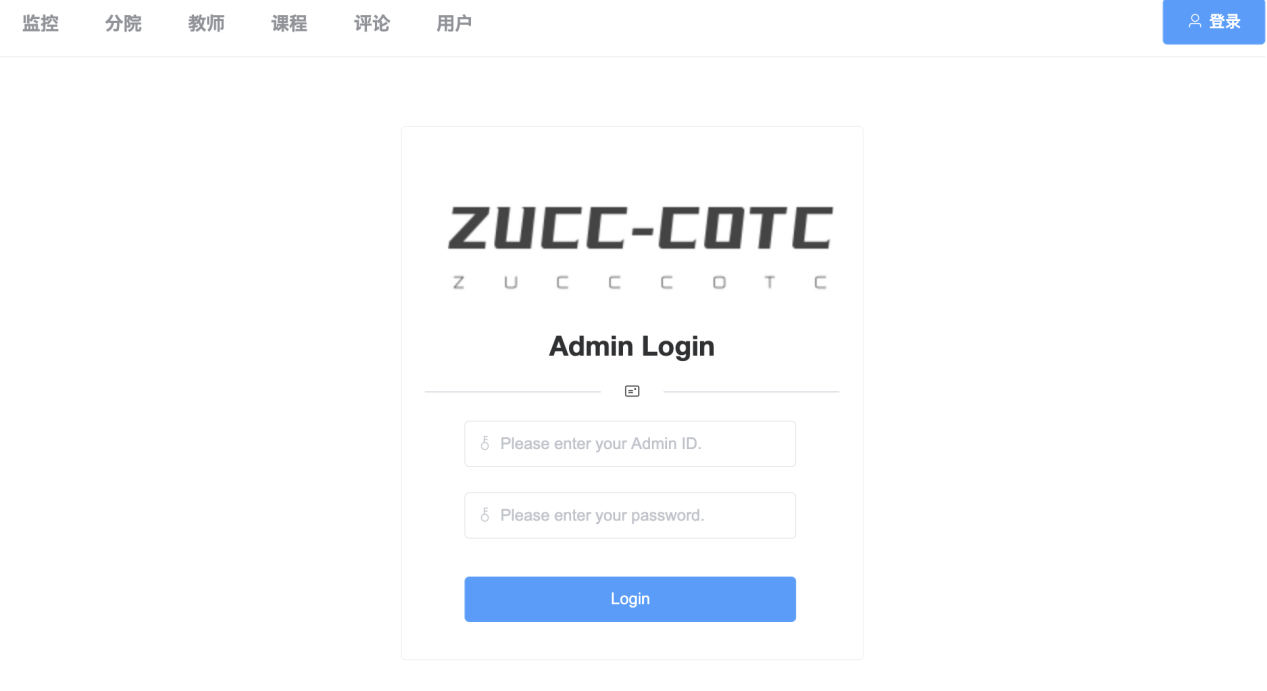
**图3.1.2.1-11 个人基础信息显示界面**



**图3.1.2.1-12 个人密码信息修改界面**



**图3.1.2.1-13-2 个人信息修改界面**

****

**图3.1.2.1-13 管理员用户登录页**

****

**图3.1.2.1-14 管理员用户管理分院内容页**

****

**图3.1.2.1-15 分院内容详情页**

****

**图3.1.2.1-16 管理员用户管理教师内容页**

****

**图3.1.2.1-17 教师内容详情页**

****

**图3.1.2.1-18 管理员用户管理课程内容页**

****

**图3.1.2.1-19 课程内容详情页**

****

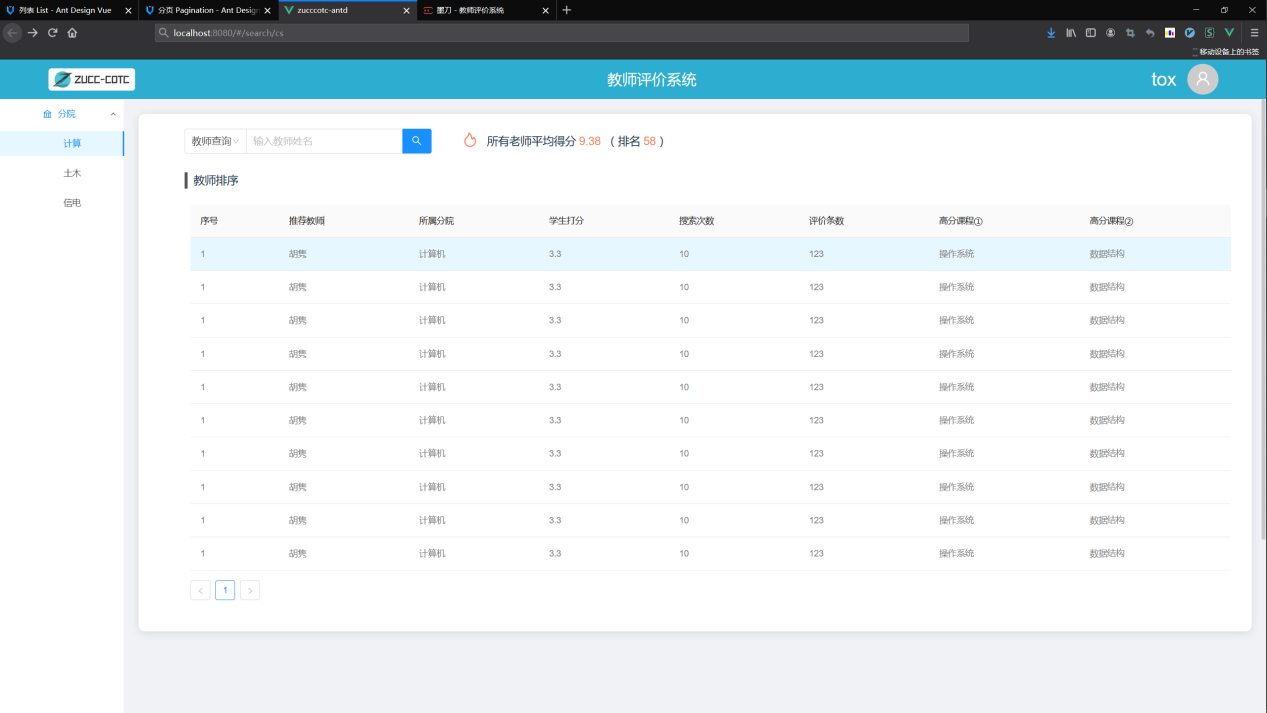
**图3.1.2.1-20 管理员用户管理评论内容页**

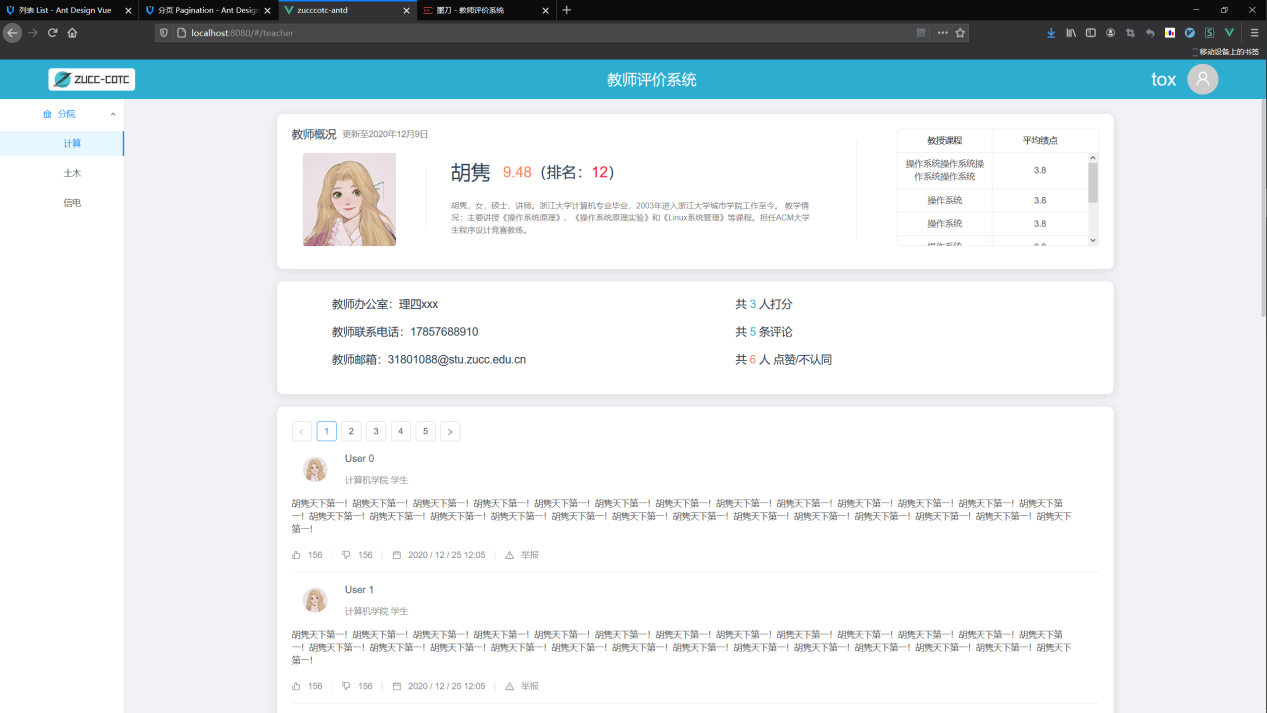
****

**图3.1.2.1-21 评论内容详情页**

**具体实现**

****

****

****

### 3.1.2.2界面原型图说明

* 图3.1.2.1-1：

|  |  |
| --- | --- |
| 名称 | 用户登录界面 |
| 内容 | 注册/登录按钮、忘记密码超链接、账号密码输入框 |
| 实现方式 | 用户初次登录使用选课网账号进行登录，后续使用自己修改密码后的账号登录时忘记密码时可以点击忘记密码超链接（还未详细做）帮助找回密码 |

* 图3.1.2.1-2：

|  |  |
| --- | --- |
| 名称 | 搜索主界面 |
| 内容 | 搜索范围选择框、分院选择栏、教师推荐模块（默认根据学生打分从高到低来）、头部栏、底部栏、老师平均得分显示、页面跳转框 |
| 实现方式 | 用户可以选择分院来进入相应的分院教师页面；选择框有教师查询、课程查询、排名查询等三方面来进行检索；平均分给学生一个老师得分高低的参考。  在教师推荐模块中，学生打分、搜索次数、评价次数可选择逆序或者顺序来排，并给出老师给学生打的绩点平均得分最高的两门课程以供参考。 |

* 图3.1.2.1-3：

|  |  |
| --- | --- |
| 名称 | 分院教师搜索界面 |
| 内容 | 在搜索主界面的基础上添加分院logo |
| 实现方式 | 具体信息部分由原来的全院教师范围改成了分院教师范围，包括具体教师罗列以及教师得分排名（都只计算相关分院） |

* 图3.1.2.1-4：

|  |  |
| --- | --- |
| 名称 | 分院——课程搜索界面 |
| 内容 | Logo+课程查询模糊搜索框+相关检索到的课程显示 |
| 实现方式 | 课程搜索结果显示在相关模块中，通过教师查询按钮进入相应的教师选择页面 |

* 图3.1.2.1-5：

|  |  |
| --- | --- |
| 名称 | 分院——课程教师搜索界面 |
| 内容 | 分院——课程搜索界面基础上加入了教师选择框+回退按钮 |
| 实现方式 | 回退按钮返回上一页面状态，教师选择框进行相应教师选择并进入该教师详情介绍页面 |

* 图3.1.2.1-5：

|  |  |
| --- | --- |
| 名称 | 教师详情介绍页面 |
| 内容 | 分院选择栏、教师概况模块、教师联系方式模块、教师评价模块、立即评价栏 |
| 实现方式 | 教师概况中显示教师大致信息、所授课程及其相应学生得分的平均绩点、学生打分得分及其得分排名；  学生若想联系该教师可以从教师联系方式模块中获取该教师办公室、联系电话、邮箱等信息；  评论模块中学生课根据最新、最热两种排序条件来查看，并可通过关键词搜索查询自己感兴趣方面的教师评价；具体评价栏中显示评论者XXX评价时间，并提供举报机制供学生反馈一些及其不合理的评价；还有点赞与拉踩机制来反映其他学生对该评论的认同程度，提高互动性与客观性，更具参考价值 |

* 图3.1.2.1-7：

|  |  |
| --- | --- |
| 名称 | 学生评价页面 |
| 内容 | 在教师详情介绍页面或教师--课程对应界面的基础上添加评价模块框 |
| 实现方式 | 通过基础页面的提交评价按钮进入该页面，学生可以选择教师是否签到及星星（类似豆瓣）来进行打分，下面是具体的评论框（可填可不填），默认提供评价要素参考，并给出关键词可供学生进行选择——快速评论。 |

* 图3.1.2.1-8：

|  |  |
| --- | --- |
| 名称 | 课程——教师选择页面 |
| 内容 | 在基础页面添加对应课程相关教授教师选择模块框 |
| 实现方式 | 该页面列举了该门课程所有教授老师，并给出了他们所教授学生所得的平均绩点供用户进行参考，点击老师则进入了相应的教师——课程详情页面 |

* 图3.1.2.1-9：

|  |  |
| --- | --- |
| 名称 | 教师——课程详情页面 |
| 内容 | 与教师详情介绍页类似 ，多了该具体课程的上课时间及点名概率 |
| 实现方式 | 学生可以根据相应上课时间来判断课表是否冲突或者是否为自己心仪的上课时间 |

* 图3.1.2.1-10：

|  |  |
| --- | --- |
| 名称 | 历史记录查询页面 |
| 内容 | 查询模块与历史评价模块 |
| 实现方式 | 可根据时间段或者教师、课程来搜索历史评价记录，点击放大镜可以进入关键词搜索  历史评价标头是评价相关课程、教师提示（若是教师页面则不显示课程），昵称是评价时所取昵称，以及评价时间，并可选择是否删除该评价 |

* 图3.1.2.1-11：

|  |  |
| --- | --- |
| 名称 | 个人基础信息修改页面 |
| 内容 | 基础信息修改模块及密码修改模块，修改密码按钮，确定修改信息按钮 |
| 实现方式 | 手机绑定需要手机验证码，点击修改密码按钮进入密码修改界面；修改其余信息则要点击确定修改信息按钮 |

* 图3.1.2.1-12：

|  |  |
| --- | --- |
| 名称 | 个人密码信息修改页面 |
| 内容 | 在个人基础信息修改页面的基础上加入具体密码修改部分 |
| 实现方式 | 密码修改处进行手机验证，更好地进行个人账号保护，并点击修改密码按钮保存密码修改信息 |

* 图3.1.2.1-13：

|  |  |
| --- | --- |
| 名称 | 管理员用户登录页面 |
| 内容 | 头部栏+账号及密码输入框+logo+用户头像+login按钮 |
| 实现方式 | 头部栏进行功能分类，管理员用户输入自己的账号进入界面 |

* 图3.1.2.1-14：

|  |  |
| --- | --- |
| 名称 | 管理员用户管理内容页面 |
| 内容 | 头部栏+账号及密码输入框+logo+用户头像+login按钮 |
| 实现方式 | 头部栏进行功能模块选择，主要以添加相关信息、查询所需信息为主 |

* 图3.1.2.1-15：

|  |  |
| --- | --- |
| 名称 | 管理员用户管理分院内容页面 |
| 内容 | 头部栏+分院输入框+确定按钮 |
| 实现方式 | 头部栏进行功能模块选择，以实现分院各类信息的添加 |

* 图3.1.2.1-16：

|  |  |
| --- | --- |
| 名称 | 管理员用户管理分院内容页面 |
| 内容 | 头部栏+分院数据+添加按钮 |
| 实现方式 | 头部栏进行功能模块选择，以查看分院各类信息 |

* 图3.1.2.1-17：

|  |  |
| --- | --- |
| 名称 | 管理员用户管理教师内容页面 |
| 内容 | 头部栏+教师输入框+确定按钮 |
| 实现方式 | 头部栏进行功能模块选择，以实现教师各类信息的添加 |

* 图3.1.2.1-18：

|  |  |
| --- | --- |
| 名称 | 管理员用户管理教师内容页面 |
| 内容 | 头部栏+教师数据+添加按钮 |
| 实现方式 | 头部栏进行功能模块选择，以查看教师各类信息 |

* 图3.1.2.1-19：

|  |  |
| --- | --- |
| 名称 | 管理员用户管理课程内容页面 |
| 内容 | 头部栏+课程输入框+确定按钮 |
| 实现方式 | 头部栏进行功能模块选择，添加课程框中输入课程名、输入课程编号、选择所属分院，以实现课程各类信息的添加 |

* 图3.1.2.1-20：

|  |  |
| --- | --- |
| 名称 | 管理员用户管理课程内容页面 |
| 内容 | 头部栏+课程数据+添加按钮 |
| 实现方式 | 头部栏进行功能模块选择，查看用户评论信息（用户信息及评论内容），点击评论内容能跳出详情框，易于进行评论管理（如通过别人的举报查看是否为违规评论并进行处理） |

* 图3.1.2.1-21：

|  |  |
| --- | --- |
| 名称 | 管理员用户管理评论内容页面 |
| 内容 | 头部栏+敏感评论选择框+确定按钮 |
| 实现方式 | 头部栏进行功能模块选择，以实现评论的各类信息的添加 |

* 图3.1.2.1-22：

|  |  |
| --- | --- |
| 名称 | 管理员用户管理用户内容页面 |
| 内容 | 头部栏+用户数据+刷新 |
| 实现方式 | 头部栏进行功能模块选择，罗列各个用户信息，后续会添加检索特定用户的功能 |

### 3.1.3硬件接口

* 服务器：

1. 指令集：x86\_64
2. CPU：1核心
3. 内存：2G
4. 带宽：2M

* 开发机1：

1. 指令集：x86\_64
2. CPU：6核心（i7)
3. 内存：16G
4. 带宽：100M

* 开发机2：

1. 指令集：x86\_64
2. CPU：4核心（i7-6700k)
3. 内存：8G
4. 带宽：100M

* 开发机3：

1. 指令集：x86\_64
2. CPU：8核心（i7-7700)
3. 内存：8G
4. 带宽：100M

### 3.1.4软件接口

1. 数据库管理系统：

* 名称：mysql
* 版本号：Ver 8.0.21 for Linux on x86\_64 (MySQL Community Server - GPL)
* 来源：清华大学开源软件镜像站

1. 分布式发布订阅消息系统：

* 名称：Kafka
* 版本号：1.0.0
* 来源：https://kafka.apachecn.org/downloads.html
* 用途：用于服务端用户行为日志采集

1. 数据库管理工具：

* 名称：mysqlworkbench
* 版本号：Ver 8.0.20 16114288 CE ( 64 bits ) Community
* 来源：dev.mysql.com/downloads/

1. Java IDE:

* 名称：Intellij Idea
* 版本号：Ultimate 2020.2
* 来源：[www.jetbrain.com](http://www.jetbrain.com)

### 3.1.5通信接口

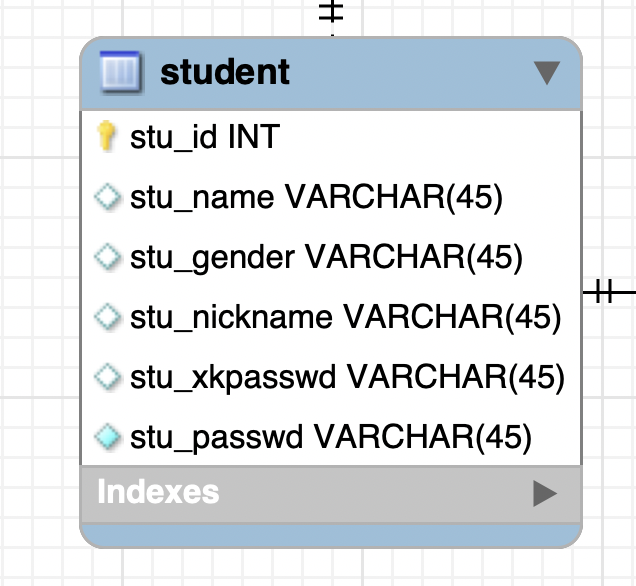
*TBD & TBDLT*

## 3.2功能需求

### 3.2.1信息流

#### 3.2.1.1数据流图-注册

##### 3.2.1.1.1数据实体



**图3.2.1 学生数据实体**

##### 3.2.1.1.2有关过程

用户在网页填写相关信息的表单，在前端验证信息合法性，若有非法字段，进行提示，若合法，发送请求到服务器，服务器判断能否生成对应账户，若可以生成对应账户，则记录到数据库，其中密码为加密后密文，若无法生成则发送response表示注册失败，则在网页上提示用户

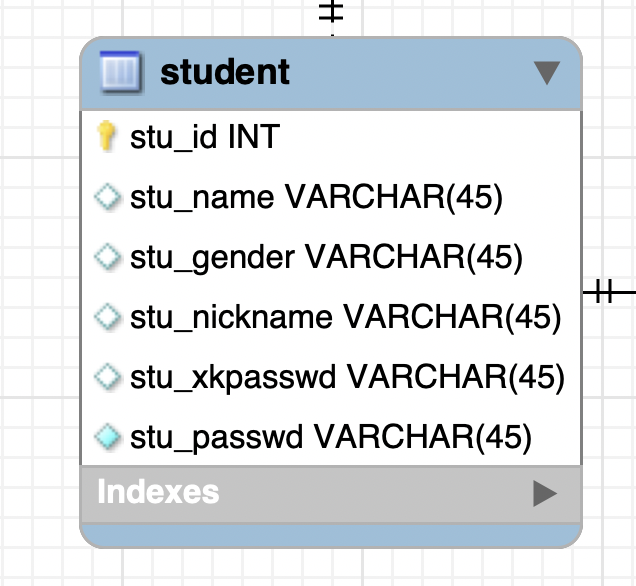
##### 3.2.1.1.3拓扑图

#### 

**图3.2.2 用户注册拓扑图**

#### 3.2.1.2数据流图-登陆

##### 3.2.1.2.1数据实体

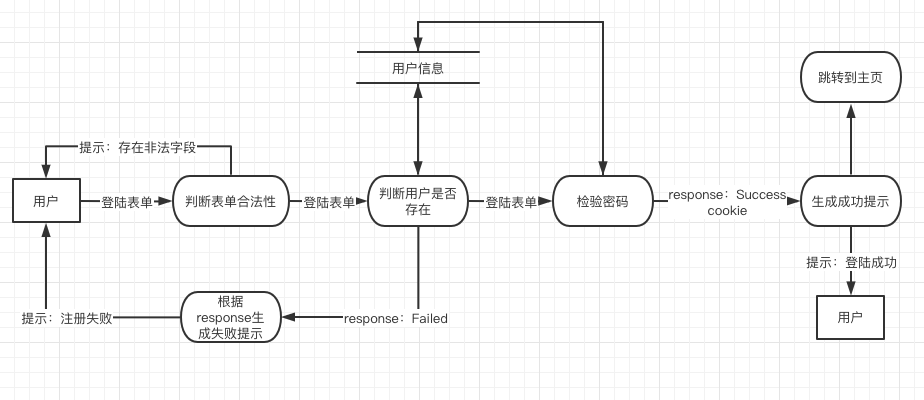


**图3.2.3（同图3.2.1） 学生用户实体**

##### 3.2.1.2.2有关过程

用户在网页填写用户名、密码并生成表单，在前端验证信息合法性，若有非法字段，进行提示，若合法，发送请求到服务器，服务器判断用户名密码是否正确/匹配，若存在该用户且密码正确，则返回带cookie的response，并提示用户登陆超过开始跳转到主页，若失败则提示用户用户名或密码错误

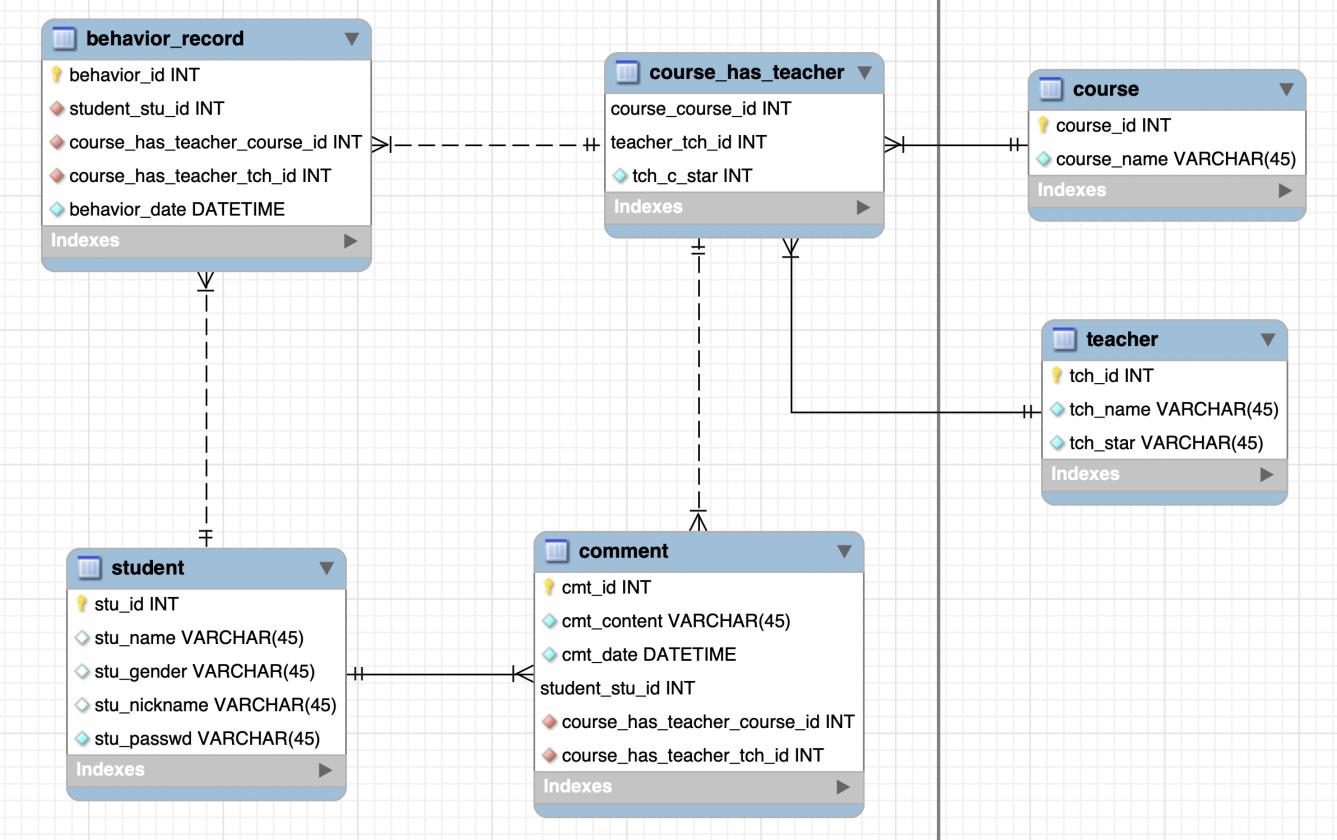
##### 3.2.1.2.3拓扑图



**图3.2.4 用户登录拓扑图**

#### 3.2.1.3数据流图-浏览

##### 3.2.1.3.1数据实体



**图3.2.5 用户浏览关系联系图**

##### 3.2.1.3.2有关过程

当用户通过搜索关键字获取条目集时：

用户在搜索框中输入关键字，然后发送给服务器；服务器收到关键字后在数据库中寻找相关信息，生成条目集返回给用户

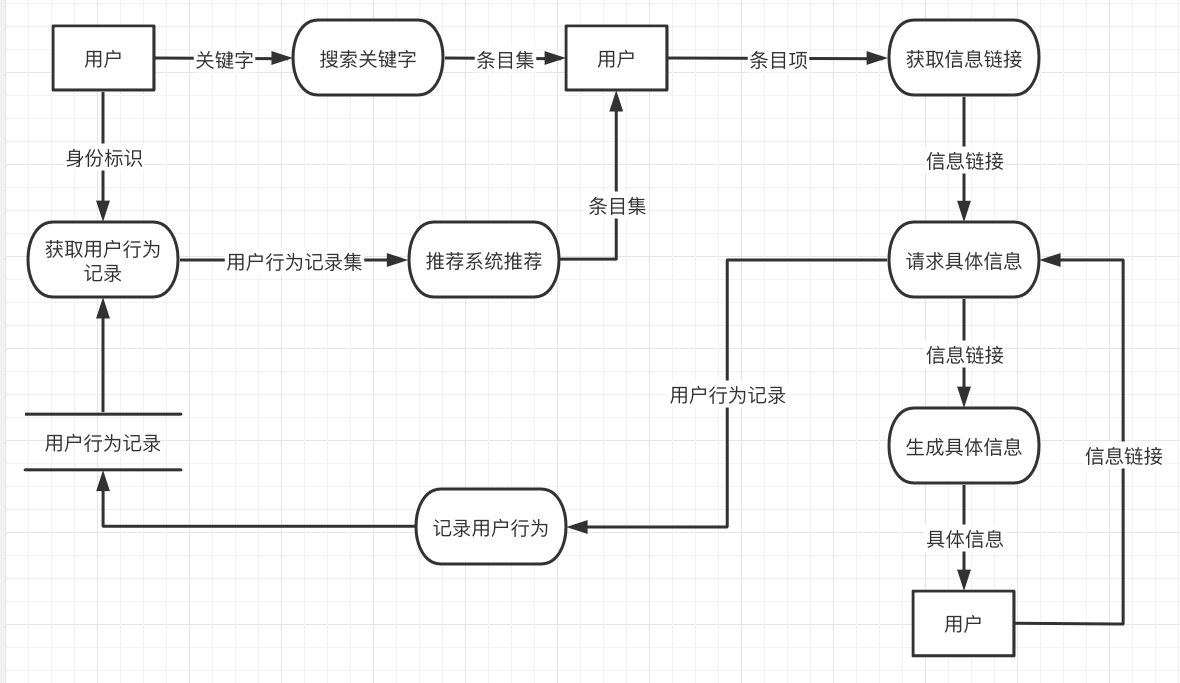
当用户通过推荐系统获取条目集时：

用户登陆后系统将获取该用户身份标示，系统根据身份标识在数据库中寻找该用户的用户行为记录，再将找到的用户行为记录打包成用户行为记录集发送给推荐系统；推荐系统根据用户行为记录集及事先计算好的标的物相似度，选出用户最有可能喜欢的几项教师/课程/教师-课程，生成条目集返回给用户

用户获取条目集后：

用户选择感兴趣的条目项，系统获取对应的信息链接以请求具体信息，服务器收到请求后生成具体信息返回给用户；用户可能在具体信息中对新的具体信息产生兴趣，于是通过对应的信息链接请求该具体信息，之后便是服务器收到请求生成具体信息返回给用户，用户在该具体信息中对信对具体信息产生兴趣发送新请求的这样一个循环

##### 3.2.1.3.3拓扑图



**图3.2.6 用户浏览拓扑图**

### 3.2.2过程描述

#### 3.2.2.1过程-注册

#### 

**表3.2.1 用户注册IPO表**

#### 3.2.2.2过程-登陆

#### 

**表3.2.2 用户登录IPO表**

#### 3.2.2.3过程-计算教师整体得分

系统：ZUCC-COTC

作者：童鑫聪

模块：评分计算

日期：2020-11-16

编号：

被调用：教师整体得分

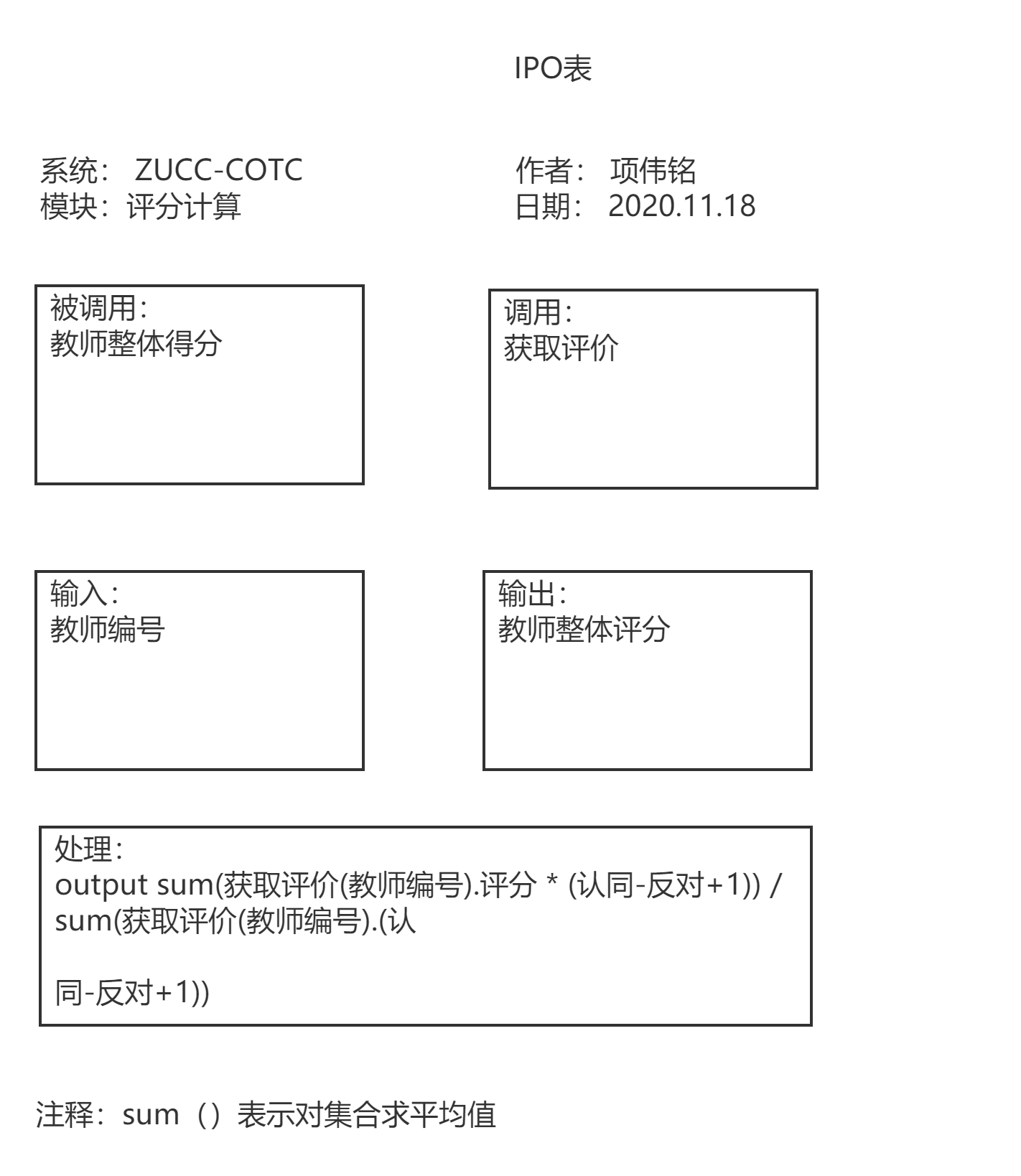
调用：获取评价

输入：教师编号

输出：教师整体评分

处理：output sum(获取评价(教师编号).评分 \* (认同-反对+1)) / sum(获取评价(教师编号).(认 同-反对+1))

注释：sum（）表示对集合求平均值



**表3.2.3 教师得分IPO表**

#### 3.2.2.4过程-计算教师点名概率

系统：ZUCC-COTC

作者：童鑫聪

模块：评价数据计算

日期：2020-11-16

编号：

被调用：点名概率

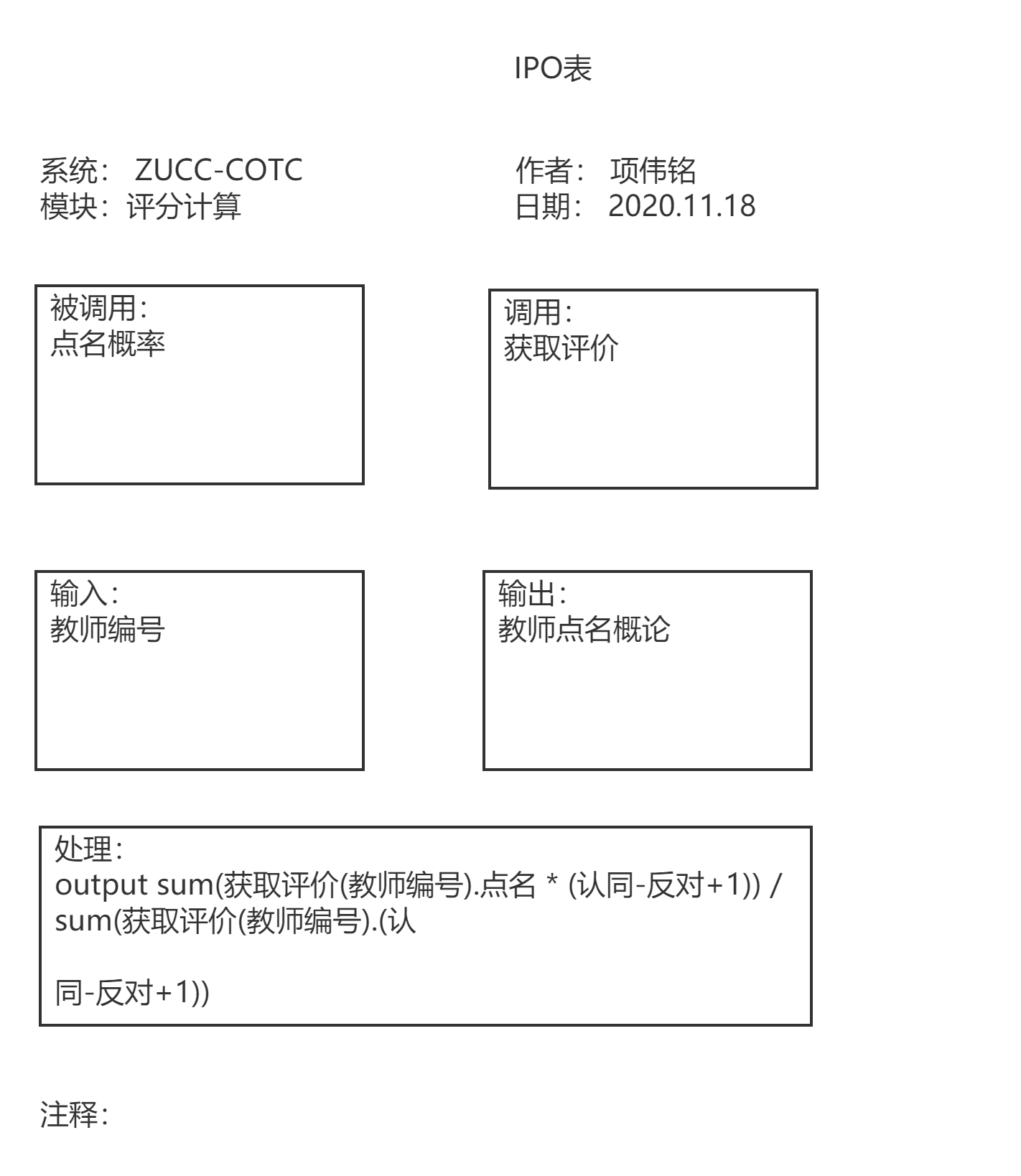
调用：获取评价

输入：教师编号

输出：教师点名概率

处理：output sum(获取评价(教师编号).点名 \* (认同-反对+1)) / sum(获取评价(教师编号).(认 同-反对+1))

注释：



**表3.2.4 教师点名概率IPO表**

#### 3.2.2.5过程-计算教师-课程平均得分

系统：ZUCC-COTC

作者：童鑫聪

模块：评价数据计算

日期：2020-11-16

编号：

被调用：点名概率

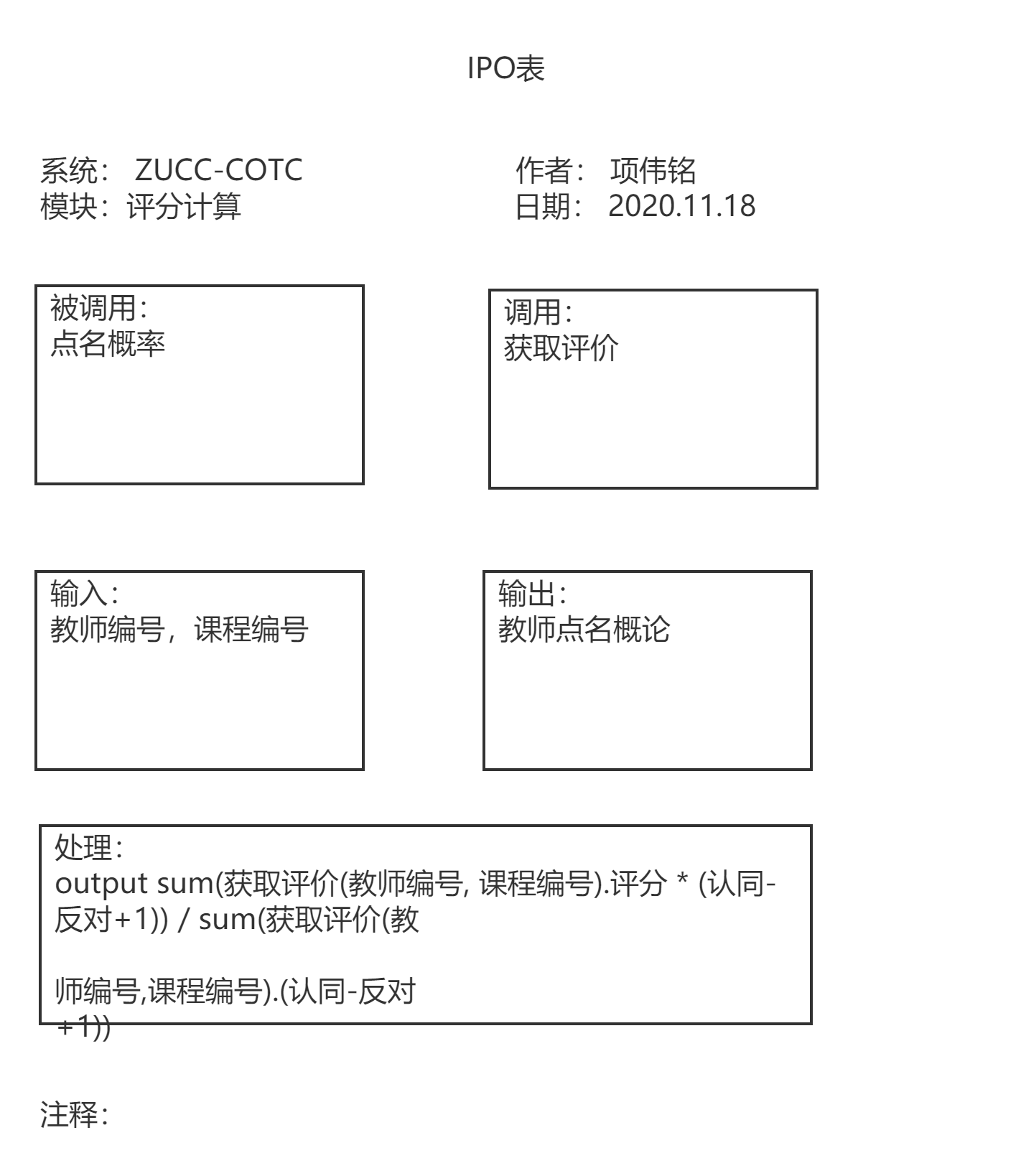
调用：获取评价

输入：教师编号，课程编号

输出：课程平均得分概率

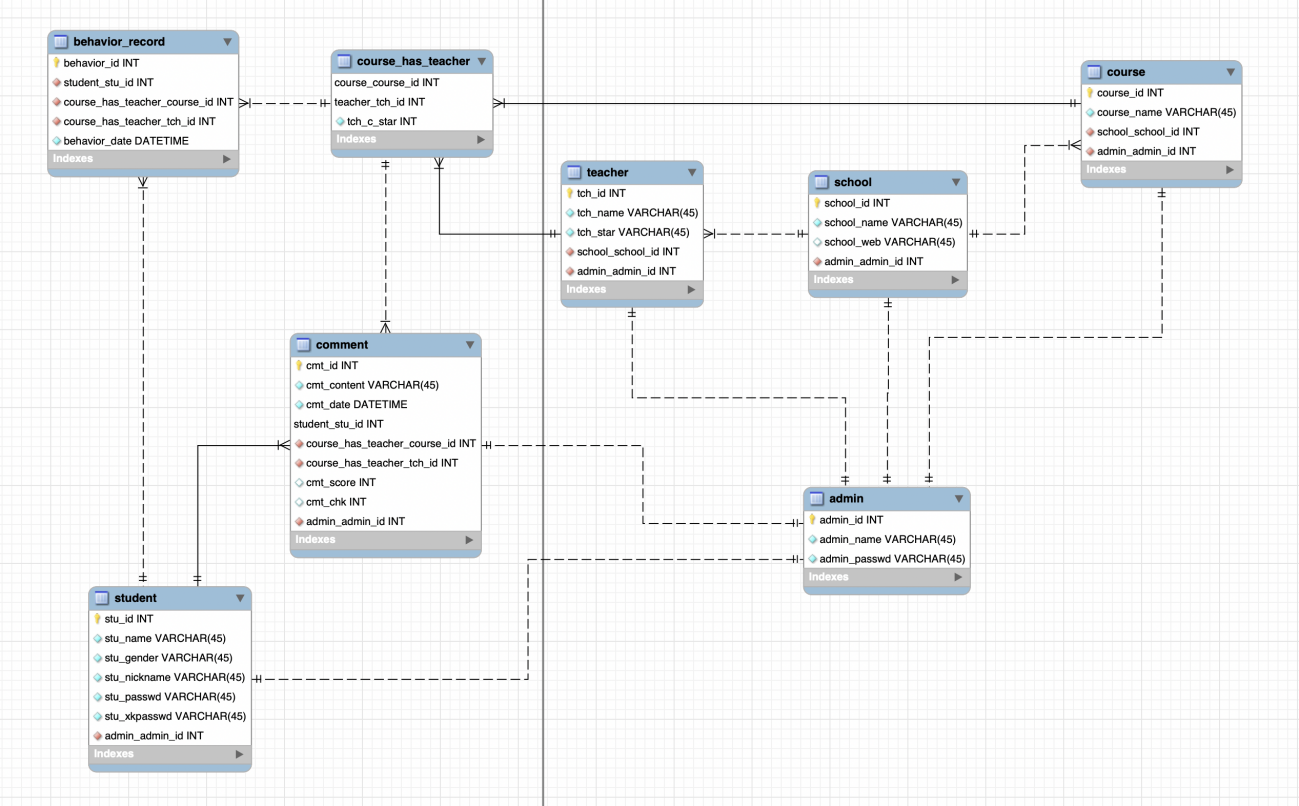
处理：output sum(获取评价(教师编号, 课程编号).评分 \* (认同-反对+1)) / sum(获取评价(教 师编号,课程编号).(认同-反对+1))

注释：



**表3.2.5 教师总分IPO表**

**3.2.3数据构建规范**



**图3.2.7 整体实体联系图**

-- MySQL Script generated by MySQL Workbench

-- Thu Nov 12 05:27:16 2020

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;

USE `mydb` ;

-- -----------------------------------------------------

-- Table `mydb`.`student`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`student` (

`stu\_id` INT NOT NULL,

`stu\_name` VARCHAR(45) NULL,

`stu\_gender` VARCHAR(45) NULL,

`stu\_nickname` VARCHAR(45) NULL,

`stu\_xkpasswd` VARCHAR(45) NULL,

`stu\_passwd` VARCHAR(45) NOT NULL,

PRIMARY KEY (`stu\_id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`teacher`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`teacher` (

`tch\_id` INT NOT NULL,

`tch\_name` VARCHAR(45) NOT NULL,

`tch\_star` VARCHAR(45) NOT NULL,

PRIMARY KEY (`tch\_id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`course`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`course` (

`course\_id` INT NOT NULL,

`course\_name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`course\_id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`course\_has\_teacher`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`course\_has\_teacher` (

`course\_course\_id` INT NOT NULL,

`teacher\_tch\_id` INT NOT NULL,

`tch\_c\_star` INT NOT NULL,

PRIMARY KEY (`course\_course\_id`, `teacher\_tch\_id`),

INDEX `fk\_course\_has\_teacher\_teacher1\_idx` (`teacher\_tch\_id` ASC) VISIBLE,

INDEX `fk\_course\_has\_teacher\_course\_idx` (`course\_course\_id` ASC) VISIBLE,

CONSTRAINT `fk\_course\_has\_teacher\_course`

FOREIGN KEY (`course\_course\_id`)

REFERENCES `mydb`.`course` (`course\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_course\_has\_teacher\_teacher1`

FOREIGN KEY (`teacher\_tch\_id`)

REFERENCES `mydb`.`teacher` (`tch\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`comment`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`comment` (

`cmt\_id` INT NOT NULL,

`cmt\_content` VARCHAR(45) NOT NULL,

`cmt\_date` DATETIME NOT NULL,

`student\_stu\_id` INT NOT NULL,

`course\_has\_teacher\_course\_id` INT NOT NULL,

`course\_has\_teacher\_tch\_id` INT NOT NULL,

PRIMARY KEY (`cmt\_id`, `student\_stu\_id`),

INDEX `fk\_comment\_student1\_idx` (`student\_stu\_id` ASC) VISIBLE,

INDEX `fk\_comment\_course\_has\_teacher1\_idx` (`course\_has\_teacher\_course\_id` ASC, `course\_has\_teacher\_tch\_id` ASC) VISIBLE,

CONSTRAINT `fk\_comment\_student1`

FOREIGN KEY (`student\_stu\_id`)

REFERENCES `mydb`.`student` (`stu\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_comment\_course\_has\_teacher1`

FOREIGN KEY (`course\_has\_teacher\_course\_id` , `course\_has\_teacher\_tch\_id`)

REFERENCES `mydb`.`course\_has\_teacher` (`course\_course\_id` , `teacher\_tch\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`behavior\_record`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`behavior\_record` (

`behavior\_id` INT NOT NULL AUTO\_INCREMENT,

`student\_stu\_id` INT NOT NULL,

`course\_has\_teacher\_course\_id` INT NOT NULL,

`course\_has\_teacher\_tch\_id` INT NOT NULL,

`behavior\_date` DATETIME NOT NULL,

PRIMARY KEY (`behavior\_id`),

INDEX `fk\_behavior\_record\_student1\_idx` (`student\_stu\_id` ASC) VISIBLE,

INDEX `fk\_behavior\_record\_course\_has\_teacher1\_idx` (`course\_has\_teacher\_course\_id` ASC, `course\_has\_teacher\_tch\_id` ASC) VISIBLE,

CONSTRAINT `fk\_behavior\_record\_student1`

FOREIGN KEY (`student\_stu\_id`)

REFERENCES `mydb`.`student` (`stu\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_behavior\_record\_course\_has\_teacher1`

FOREIGN KEY (`course\_has\_teacher\_course\_id` , `course\_has\_teacher\_tch\_id`)

REFERENCES `mydb`.`course\_has\_teacher` (`course\_course\_id` , `teacher\_tch\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

### 3.2.4数据字典

#### 3.2.4.1数据元素1

名称：课程评价

描述：用户对某个教师某门课程的评价

定义： 课程评价 = 评价人 + 评价内容 + 打分 + 评价时间 + 认同人数 + 否认人数

范围：

#### 3.2.4.2数据元素2

名称：教师主页信息

描述：关于某个教师，平台可呈现给用户的所有信息

定义：教师主页信息 = 教师名字 + 照片 + 评分 + 简介 + 学院 + 0{教师-课程具体信息 的信息链接}n + 0{课程评价}n

范围：

#### 3.2.4.3数据元素3

名称：课程主页信息

描述：关于某门课程，平台可呈现给用户的所有信息

定义：课程具体信息 = 课程名 + 封面 + 简介 + 0{教师主页信息的信息链 接}n + 0{课程评 价}n

范围：

#### 3.2.4.4数据元素4

名称：教师-课程具体信息

描述：关于某个教师的某门课程，平台可呈现给用户的所有信息

定义：教师-课程具体信息 = 教师名字 +（照片）+ 得分 + 0{课程评价}n

范围：

#### 3.2.4.5数据元素5

名称：信息链接

描述：指向教师具体信息、课程具体信息、教师-课程具体信息的链接

定义：信息链接 = URI

范围：

#### 3.2.4.6数据元素6

名称：教师条目项

描述：搜索、推荐结果中代表一个具体教师的一项，包含概要信息与信息链接

定义：教师条目项 = 教师名字 + 教师简介 + 教授课程清单

范围：

#### 3.2.4.7数据元素7

名称：教师-课程条目项

描述：搜索、推荐结果中代表一个教师-课程的一项，包含概要信息与信息链接

定义：教师-课程条目项 = 教师名字 + 课程名字 + 得分

范围：

#### 3.2.4.8数据元素8

名称：条目项

描述：搜索、推荐结果中代表一个具体教师、具体课程、具体教师-课程对的一项，包含概 要信息与信息链接

定义：条目项 = [ 教师条目项 | 课程条目项 | 教师-课程条目项 ]

范围：

#### 3.2.4.9数据元素9

名称：条目集

描述：搜索、推荐的条目的集合

定义：条目集 = 0{条目项}n

范围：

#### 3.2.4.10数据元素10

名称：用户行为记录

描述：用户每请求一个信息链接就产生一条用户行为记录，用于推荐系统

定义：用户行为记录 = 用户唯一身份表示 + 时间 + 信息链接

范围：

## 3.3性能需求

* 日常时段能承受500用户同时在线，且所有业务动作可用稳定，页面跳转时长不超过5秒钟；高峰时段承受1000用户同时在线，且所有业务动作可用、稳定，页面跳转时长不超过5秒
* 夜间根据用户历史记录在不影响业务的前提下2h内计算K近邻以支持协同过滤算法实现的推荐系统

## 3.4设计约束

* xk网防刷机制限制了获取用户历史选课记录的速度和即使性

## 3.5软件系统属性

#### 3.5.1可靠性

* 系统应保证7\*24小时不间断运行
* 服务器应在高峰期时实现20分钟内的配置升级
* 平均故障时间间隔应不低于2880h

#### 3.5.2可用性

**--易学性：**类似于豆瓣的机制，但本产品只有教师、课程两个维度，内容精简得多，供操作部分也相对较少，学习起来非常简易。

**--交互效率：**此处主要有两点评估，一个是系统响应操作的速度，另一个是用户对说明书或帮助功能的使用频率，响应速度受限于服务器的网络资源，而就帮助功能而言，其实并没有太多，因为某项功能其组成的操作并不多，往往一两次交互就能出来用户想要的结果。

**--易记性：** 当用户停止使用系统一段时间后重新使用本系统，因为系统本身的操作并不多，故重新使用也基本不会忘记如何使用。

--**出错频率及包容度：**出错频率和包容度指的是用户在操作过程中出现的误操作概率，以及出现错误后系统或产品的错误包容度，因为本产品是提供评论以及发表评论并在后台做大数据分析的，本身并不涉及到选课操作，只对选课提供一个参考意见，个人的一个评价在大数据面前是很无力的，因此出错对系统本身造成的影响并不大。

--**用户满意度：**用户满意度相对于前面四个指标来说，更多的是用户的主观体验和感受，建立在前面四个指标之上做出的综合评价，就像是用户角色模型是对用户研究三个方面模型的综合描述一般，以上四个维度的可用性相对较强，因此对用户满意度这一块的可用性问题并不大。

#### 3.5.3安全保密性

* 使用HTTPS协议
* 用户密码不得以明文存于数据库，应当存储以不可逆的加密算法加密后的结果
* 用户登录时需输入账户密码，密码输入三次不正确，则封号24小时
* 服务器应装有高性能的防火墙和杀毒软件以应对黑客攻击
* 限制权限，做好数据备份

## 3.6其他需求

*TBD*