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](#_Toc16223)

[1.1目的 4](#_Toc20002)

[1.1.1文档编写目的 4](#_Toc22031)

[1.1.2预期读者 4](#_Toc13080)

[1.2范围 4](#_Toc18751)

[1.2.1产品名称 4](#_Toc7154)

[1.2.2软件的应用 4](#_Toc19856)

[1.3定义、简写和缩略语 5](#_Toc16388)

[1.4引用文件 5](#_Toc1394)

[1.5综述 5](#_Toc1724)

[2总体描述 6](#_Toc22095)

[2.1产品描述 6](#_Toc21677)

[2.2产品功能 6](#_Toc7169)

[2.3用户特点 6](#_Toc8625)

[2.3.1用户调研记录 7](#_Toc21709)

[2.4约束 9](#_Toc10597)

[2.4.1法规政策 9](#_Toc29318)

[2.4.2硬件局限 9](#_Toc20565)

[2.4.3与其他应用的接口 9](#_Toc8583)

[2.4.4并行操作 9](#_Toc27520)

[2.4.5审核功能 9](#_Toc7323)

[2.4.6控制功能 9](#_Toc17704)

[2.4.7高级语言需求 10](#_Toc7442)

[2.4.8信号握手协议 10](#_Toc20752)

[2.4.9可靠性需求 10](#_Toc12435)

[2.4.10应用的关键性 10](#_Toc21382)

[2.4.11安全和保密安全考虑 10](#_Toc19050)

[2.5假设和依赖关系 10](#_Toc6643)

[2.6需求分配 10](#_Toc21608)

[3具体需求 11](#_Toc6348)

[3.1外部接口需求 11](#_Toc25480)

[3.1.1用户界面 11](#_Toc8720)

[3.1.2界面原型图 11](#_Toc5877)

[13](#_Toc16572)

[3.1.2硬件接口 14](#_Toc7218)

[3.1.3软件接口 14](#_Toc26845)

[3.1.4通信接口 15](#_Toc26981)

[3.2功能需求 15](#_Toc26915)

[3.2.1信息流 15](#_Toc1815)

[3.2.2过程描述 20](#_Toc12837)

[3.2.3数据构建规范 23](#_Toc19692)

[3.2.4数据字典 26](#_Toc26808)

[3.3性能需求 28](#_Toc20763)

[3.4设计约束 29](#_Toc9224)

[3.5软件系统属性 29](#_Toc2229)

[3.6其他需求 29](#_Toc32072)

# 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

## 课程评价是指根据一定的标准和课程系统信息以科学的方法检查课程的目标、编订和实施是否实现了教育目的，实现的程度如何，以判定[课程](https://baike.baidu.com/item/%E8%AF%BE%E7%A8%8B" \t "https://baike.baidu.com/item/%E8%AF%BE%E7%A8%8B%E8%AF%84%E4%BB%B7/_blank)设计的效果，并据此作出改进课程的决策。

为了有效监督教师教学，加强教学管理，提高教学质量，切实为学生创造良好的学习环境和平台，特此开发此有给教学质量评价平台，可供学生对其大学期间上过的课程与老师进行评价，以供后来学生选课的参考。

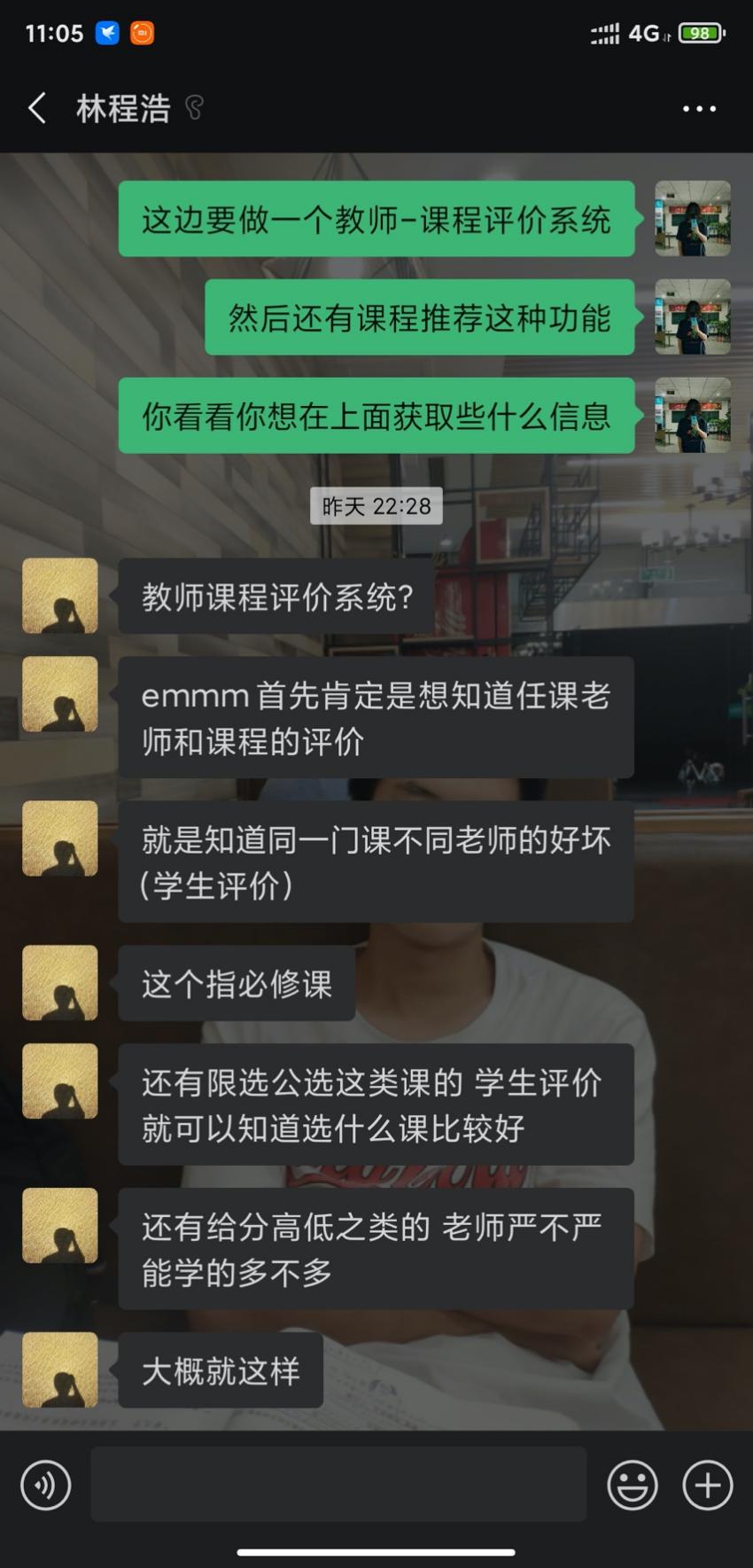
## 2.2产品功能

1. 用户可以根据教师名、课程名、课程编号搜索课程，或根据教师名搜索教师
2. 登陆用户在搜索界面可以收到系统根据该用户历史记录推荐的课程、老师
3. 用户可查看教师的整体得分、简介、任教课程、收到的评价
4. 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
5. 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、收到的评价
6. 登陆用户可对教师-课程作出评价
7. 登陆用户可对他人作出的评价表示“认同/反对”
8. 用户客队含有不当内容的言论进行举报

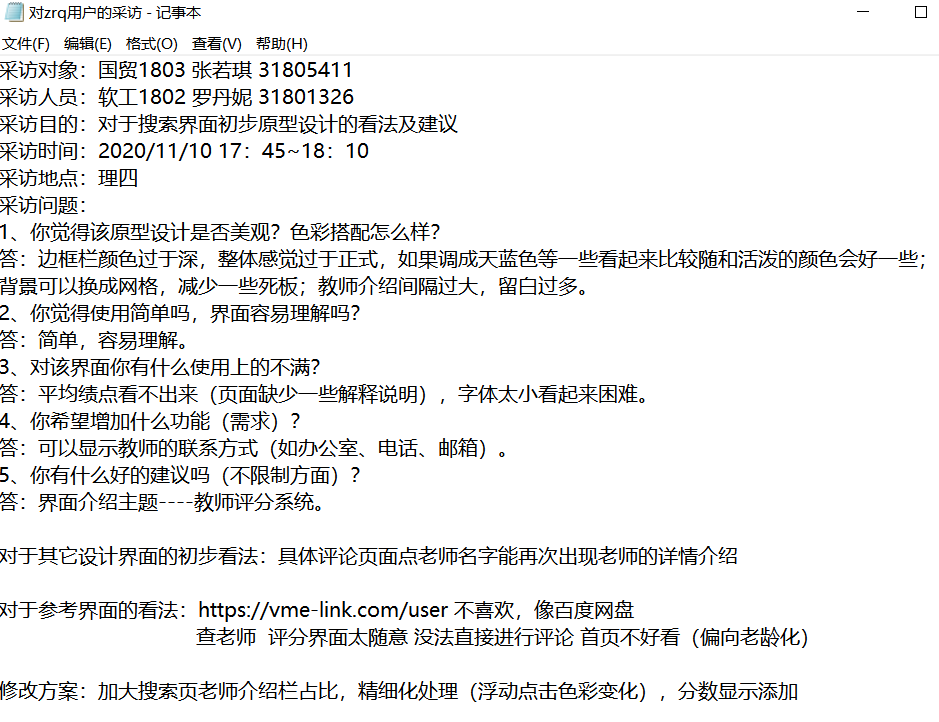
## 2.3用户特点

* 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

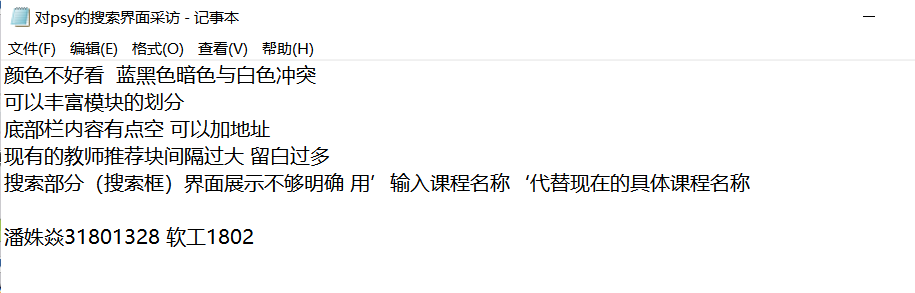
### 2.3.1用户调研记录



**图2.3.1 用户调研记录1**



**图2.3.2 用户调研记录2**



**图2.3.3 用户调研记录3**

## 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

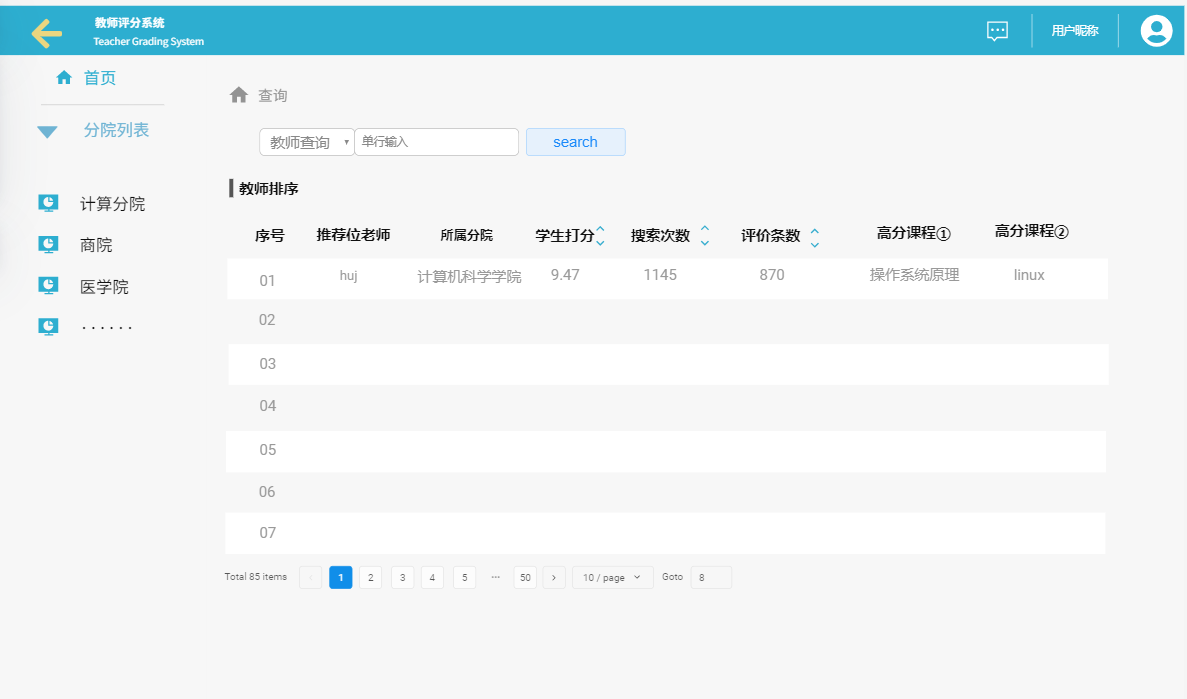
### 3.1.2界面原型图



**图3.1.1 登录界面**



**图3.1.2 特定教师界面**



**图3.1.3 搜索界面**



**图3.1.4 评论界面**

### 

**图3.1.5 提交评价界面（暂定弹出式）**

### 3.1.2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

### 3.1.3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)

### 3.1.4通信接口

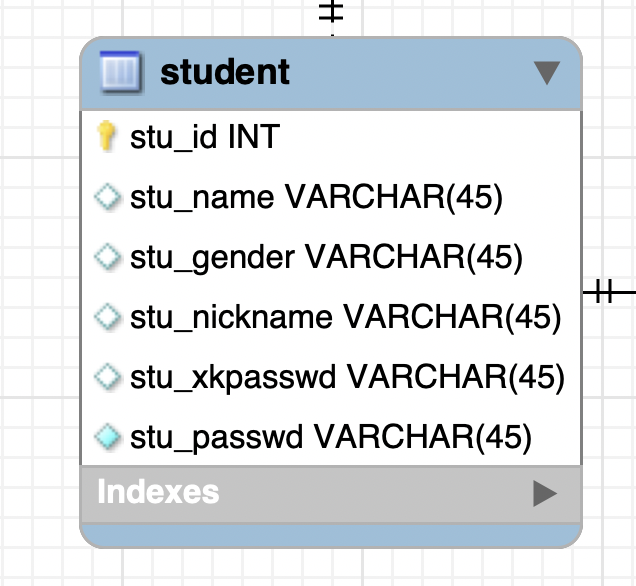
*TBD & TBDLT*

## 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

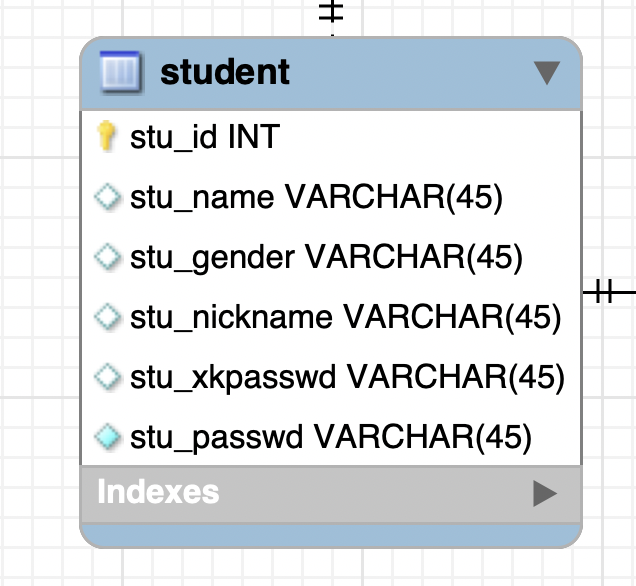
##### 3.2.1.1.3拓扑图

#### 

**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

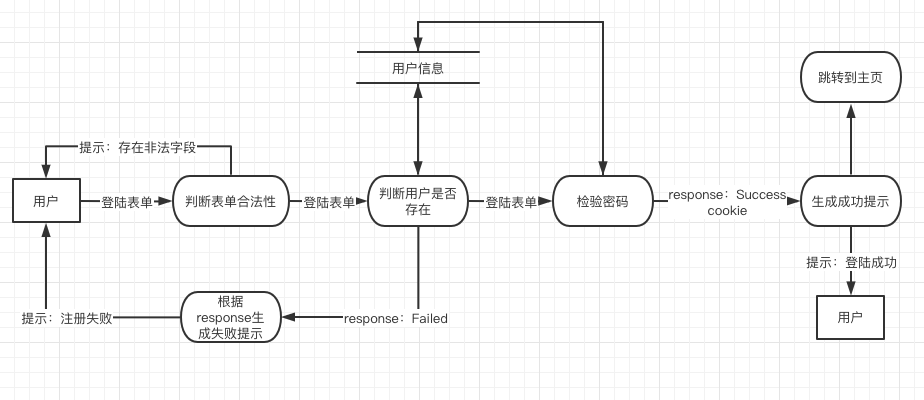


**图3.2.3（同图3.2.1） 学生用户实体**

##### 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

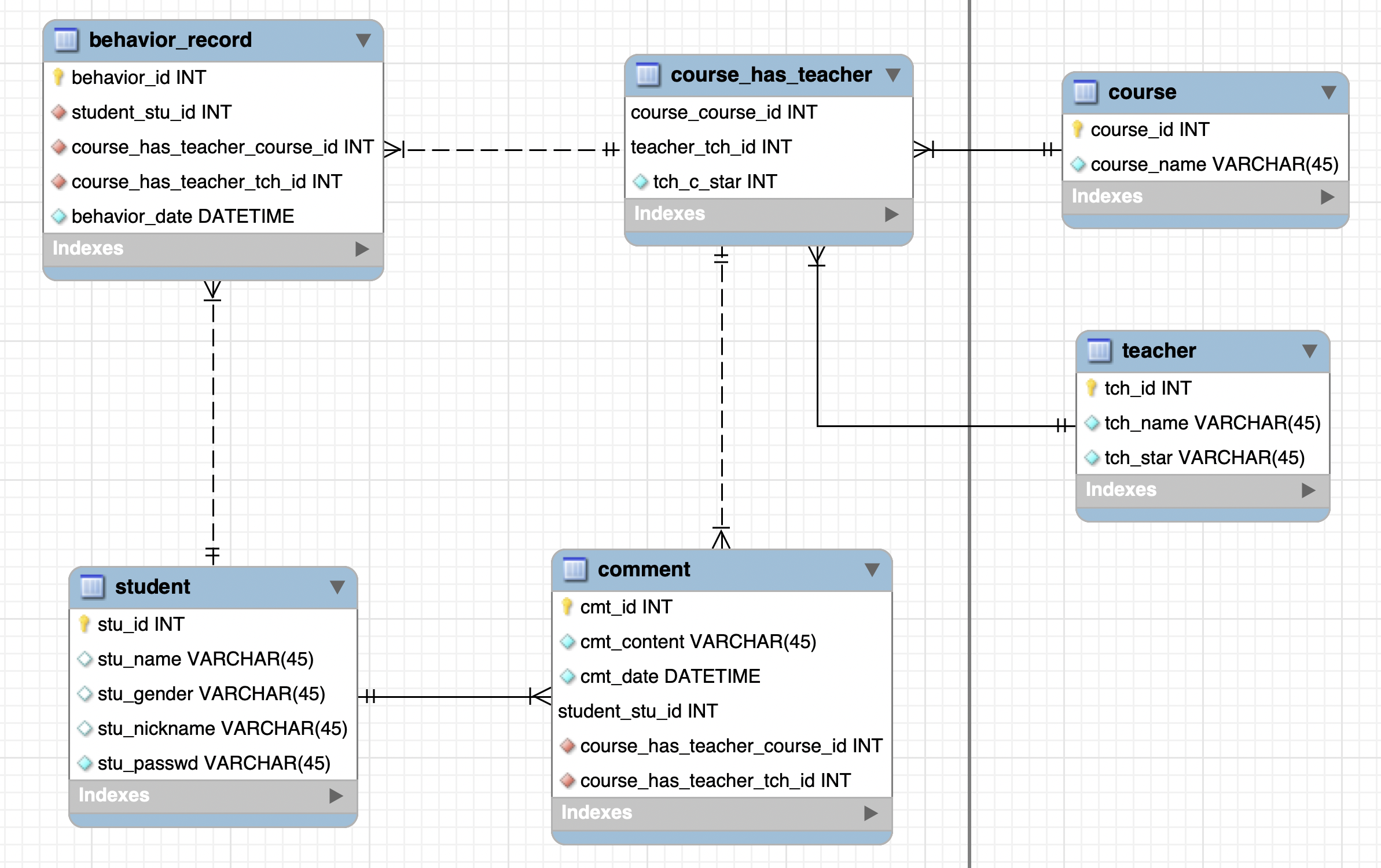
##### 3.2.1.2.3拓扑图



**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

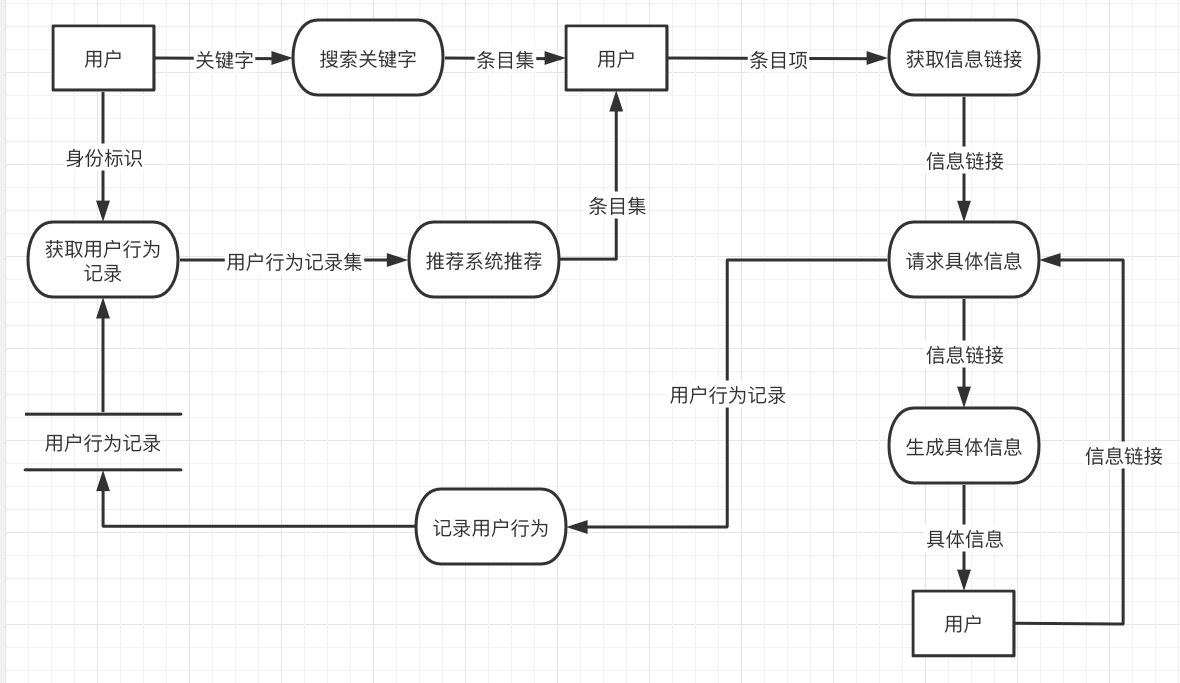
当用户通过推荐系统获取条目集时：

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

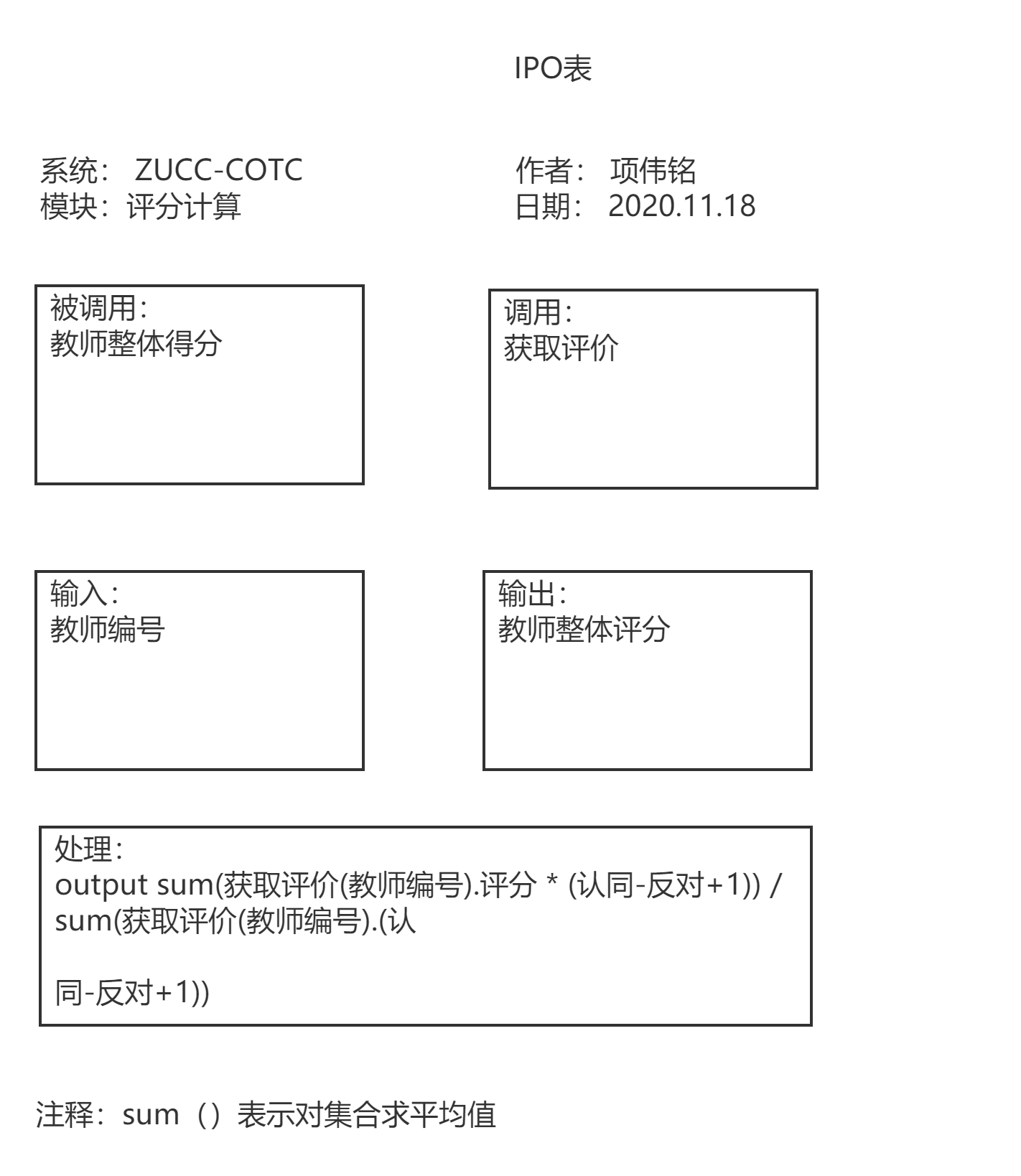
调用：获取评价

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

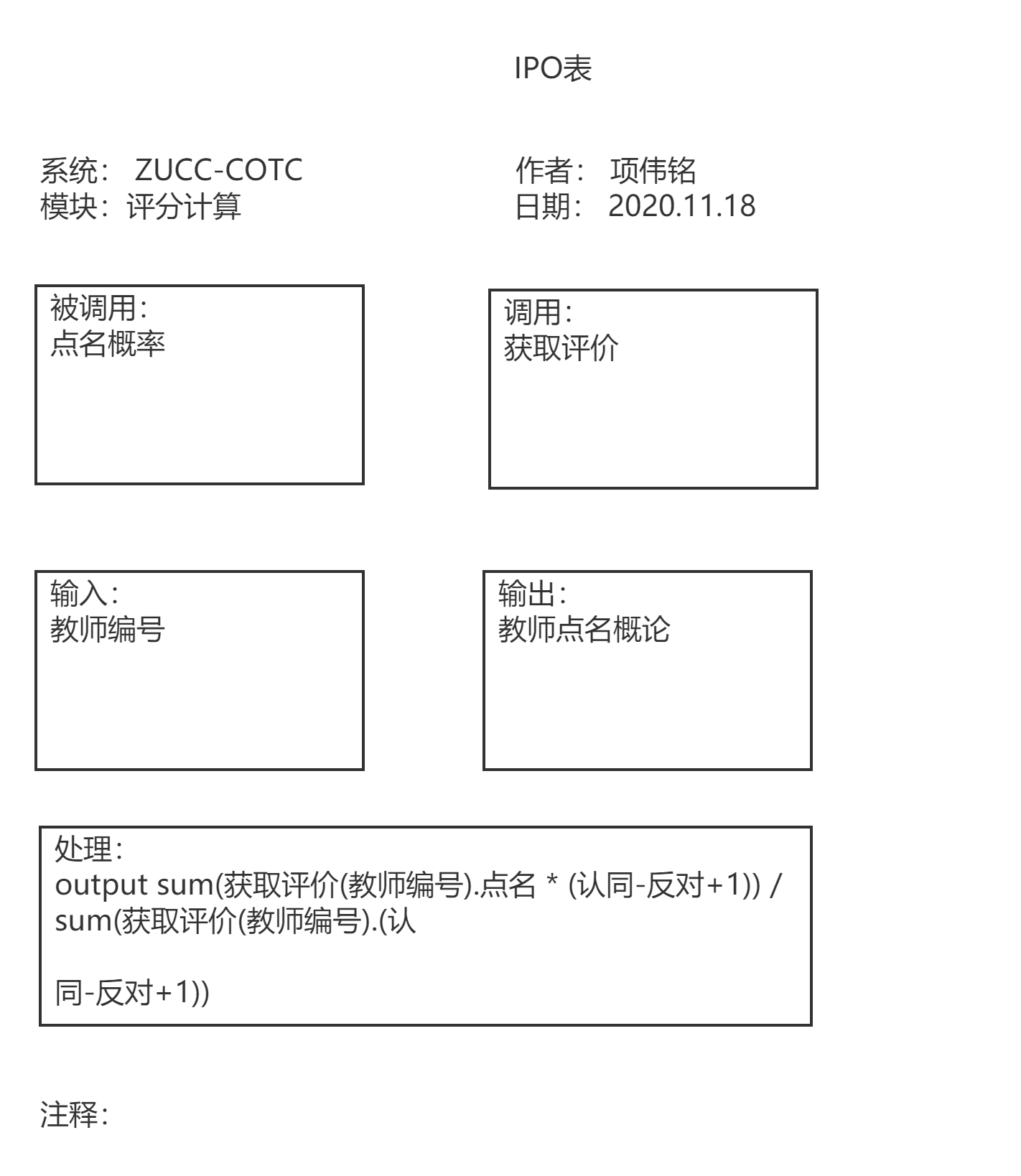
调用：获取评价

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

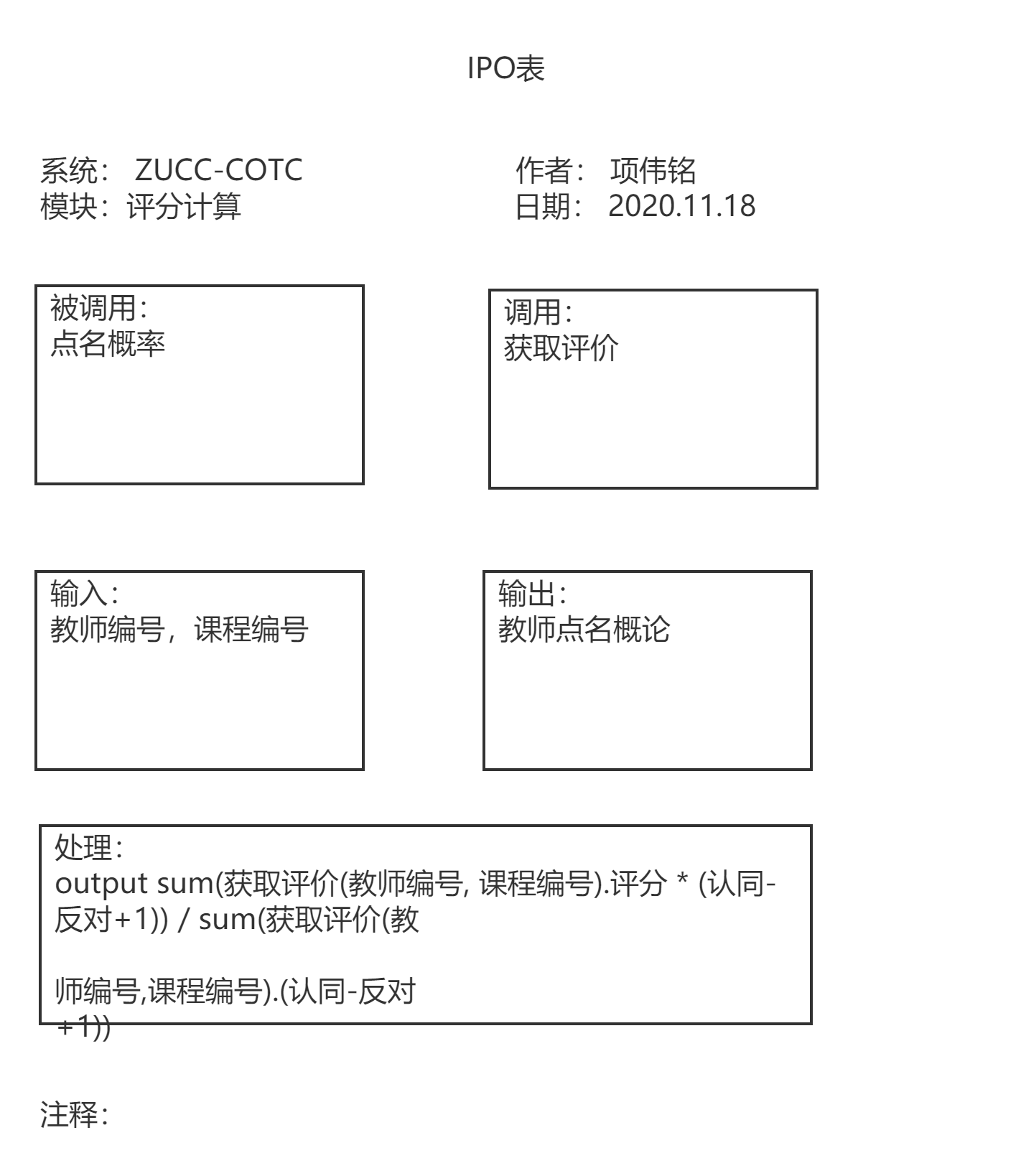
调用：获取评价

输入：教师编号，课程编号

输出：课程平均得分概率

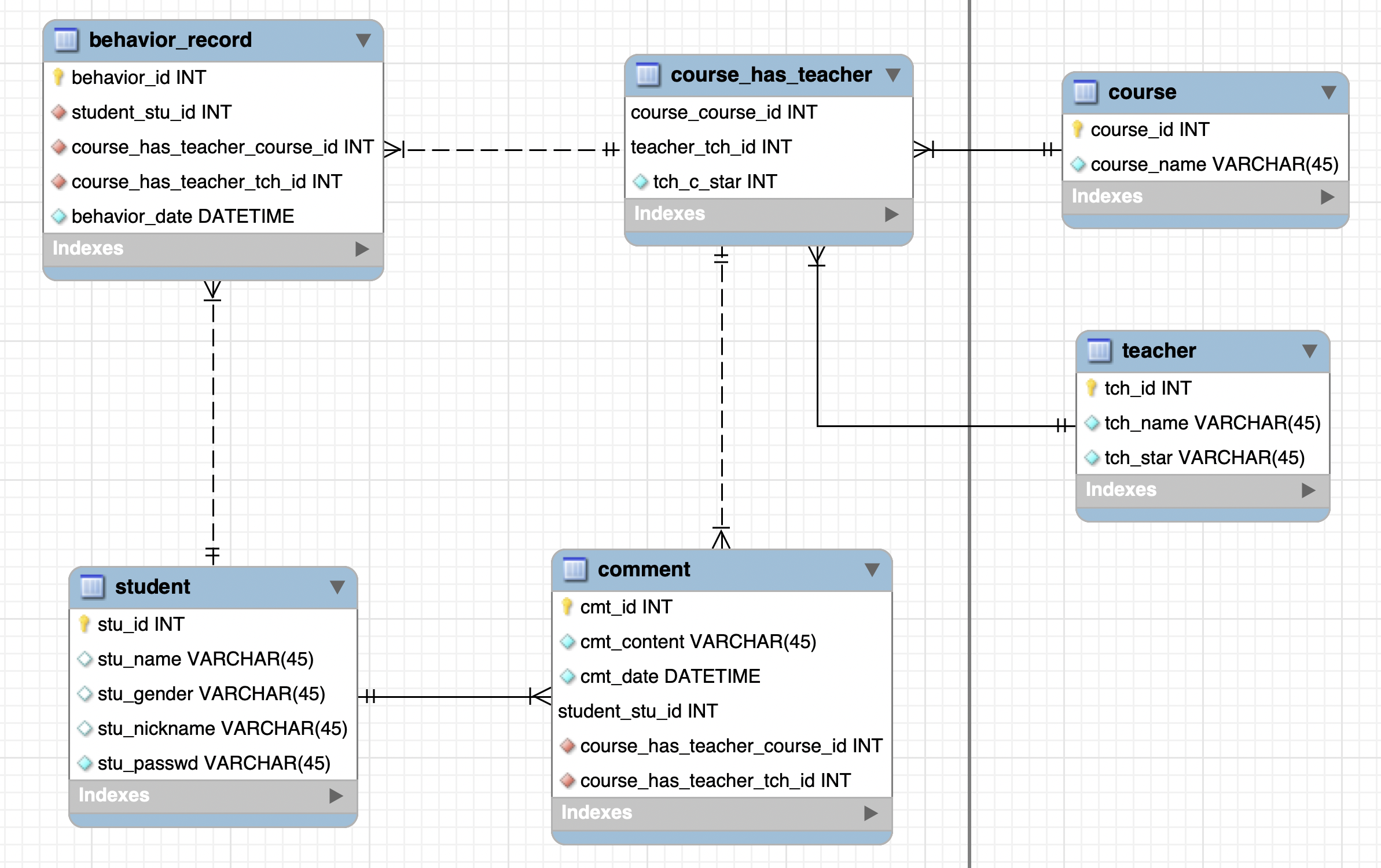
处理：output sum(获取评价(教师编号, 课程编号).评分 \* (认同-反对+1)) / sum(获取评价(教 师编号,课程编号).(认同-反对+1))

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

* 可登陆、搜索、浏览、留言
* 可使用推荐系统
* 可进行举报
* 可查询教师/教师-课程整体评分
* 可查询教师-课程平均给分
* 可查询点名概率
* 可保证在选课期间可用

#### 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*