

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于GitHub的实验管理平台的详细设计 | | |
| 课程 | 软件系统分析与设计技术 | | |
| 学 院 | 信息科学与工程学院 | | |
| 专 业 | 软件工程 | 年级 | 2017级 |
| 学生姓名 | 胡宏坤 | 学号 | 201710414305 |
| 指导教师 | 赵卫东 | 职称 | 副教授 |

|  |  |  |  |
| --- | --- | --- | --- |
| **评分项** | **评分标准** | **满分** | **得分** |
| 文档整体 | 文档内容详实、规范，美观大方 | 20 |  |
| 用例图及规约 | 用例图完整，准确，能够完全体现需求 | 10 |  |
| 顺序图与活动图 | 顺序图能够完全描述用例的设计思路和业务流程，活动图能够描述部分算法的流程。 | 10 |  |
| 类图 | 类图能够完整，准确反映业务的需数据的组织结构 | 10 |  |
| 数据库 | 数据库表的设计来自于类图，合理 | 10 |  |
| 界面设计 | 界面设计美观，清晰，合理，能够完全反映用例图的设计思路 | 10 |  |
| API接口设计 | 为每个界面设计合理的API接口，以便进行前后端分离开发 | 15 |  |
| 内容一致性 | 用例图，类图、数据库、界面相互印证，相互依赖，环环相扣 | 15 |  |
| **得分合计** | | |  |

2020 年 5 月 1 日

# **实验6：基于GitHub的实验管理平台的分析与设计**

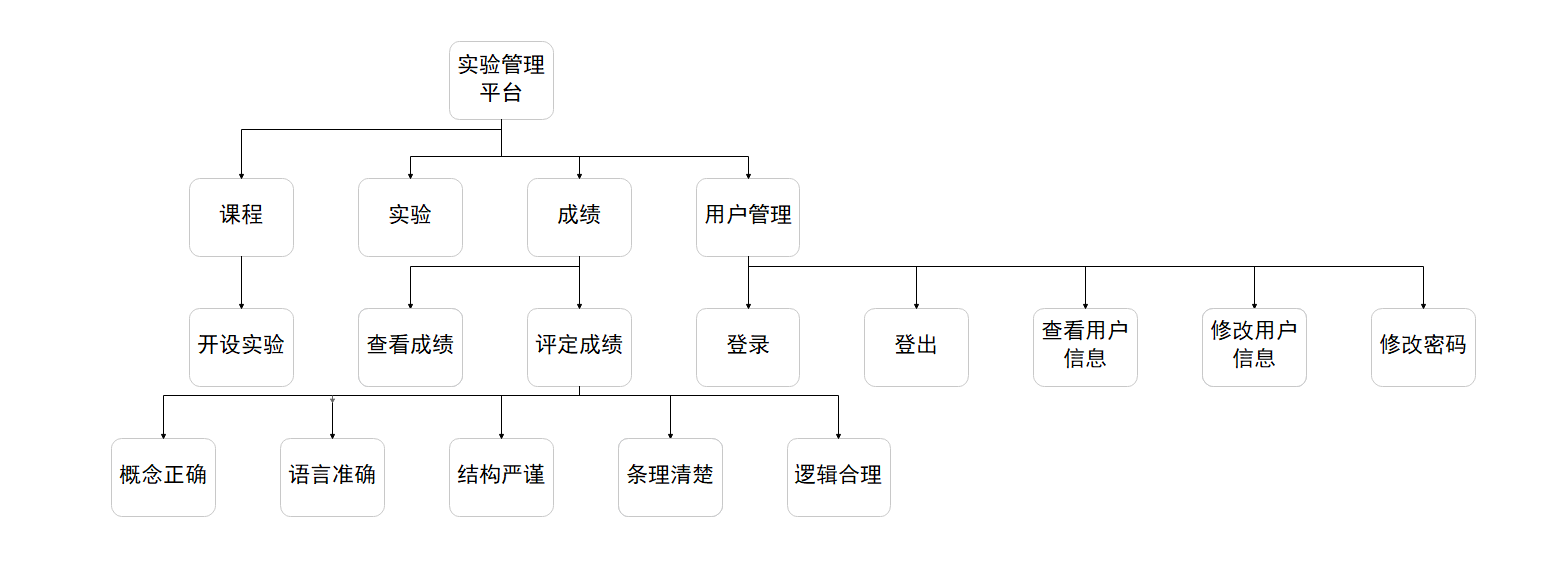
## **成都大学信息科学与工程学院**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 年级 | 专业 | 班级 | 姓名 | 学号 |
| 2017 | 软件工程 | 3班 | 胡宏坤 | 201710414305 |

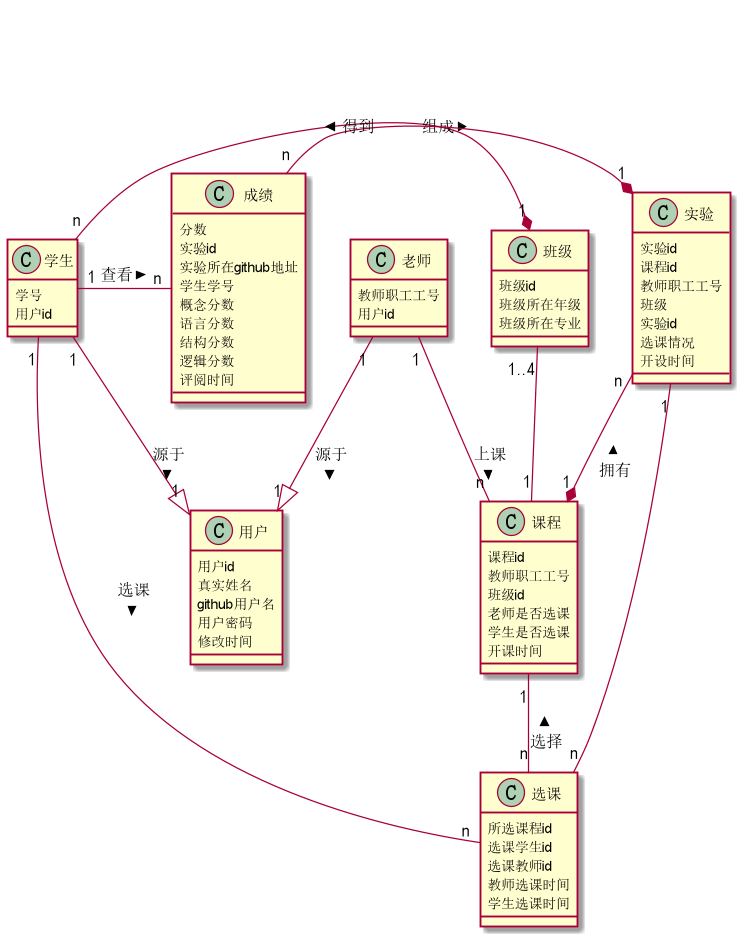
## **1. 概述**

* 基于GitHub的实验管理平台的作用是在线管理实验成绩的Web应用系统，学生和教师的实验内容均存放在GitHUB 页面上；
* 用户的功能主要有五点：
  1. 登录；
  2. 登出；
  3. 查看自己的用户信息；
  4. 修改自己的用户信息；
  5. 修改自己的密码；
* 学生的功能主要有三点：
  1. 选择课程；
  2. 设置自己的GitHub实验地址；
  3. 查询自己的成绩；
* 教师的功能主要有六点：
  1. 维护课程的信息；
  2. 维护实验的信息；
  3. 维护学生的账号；
  4. 批改每个学生的实验成绩；
  5. 查看每个学生的成绩；
* 每个课程有两种选课方式：
  1. 教师直接选择上课班级；
  2. 学生自主选课；
* 每个学生的实验的成绩分为五个个评分项：
  1. 概念是否正确，20分；
  2. 语言是否准确，20分；
  3. 结构是否严谨，20分；
  4. 条理是否清楚，20分；
  5. 逻辑是否合理，20分；
* 老师和学生都能通过本系统的链接方便地跳转到学生的每个GitHUB实验目录，以便批改实验或者查看实验情况。
* 实验成绩按数字分数计算，每项实验的满分为100分，最低为0分。
* 系统自动计算每个学生的所有实验的平均分。

## **2. 系统总体结构**



## **3. 类图设计**



类图源码 flow\_class.puml

## **4. 数据库设计**

### **4.1 用户表**

数据库建表 table\_user.md

### **4.2 教师表**

数据库建表 table\_teacher.md

### **4.3 学生表**

数据库建表 table\_student.md

### **4.4 班级表**

数据库建表 table\_class.md

### **4.5 科目表**

数据库建表 table\_subject.md

### **4.6 课程表**

数据库建表 table\_lesson.md

### **4.7 选课表**

数据库建表 table\_choose.md

### **4.8 实验表**

数据库建表 table\_test.md

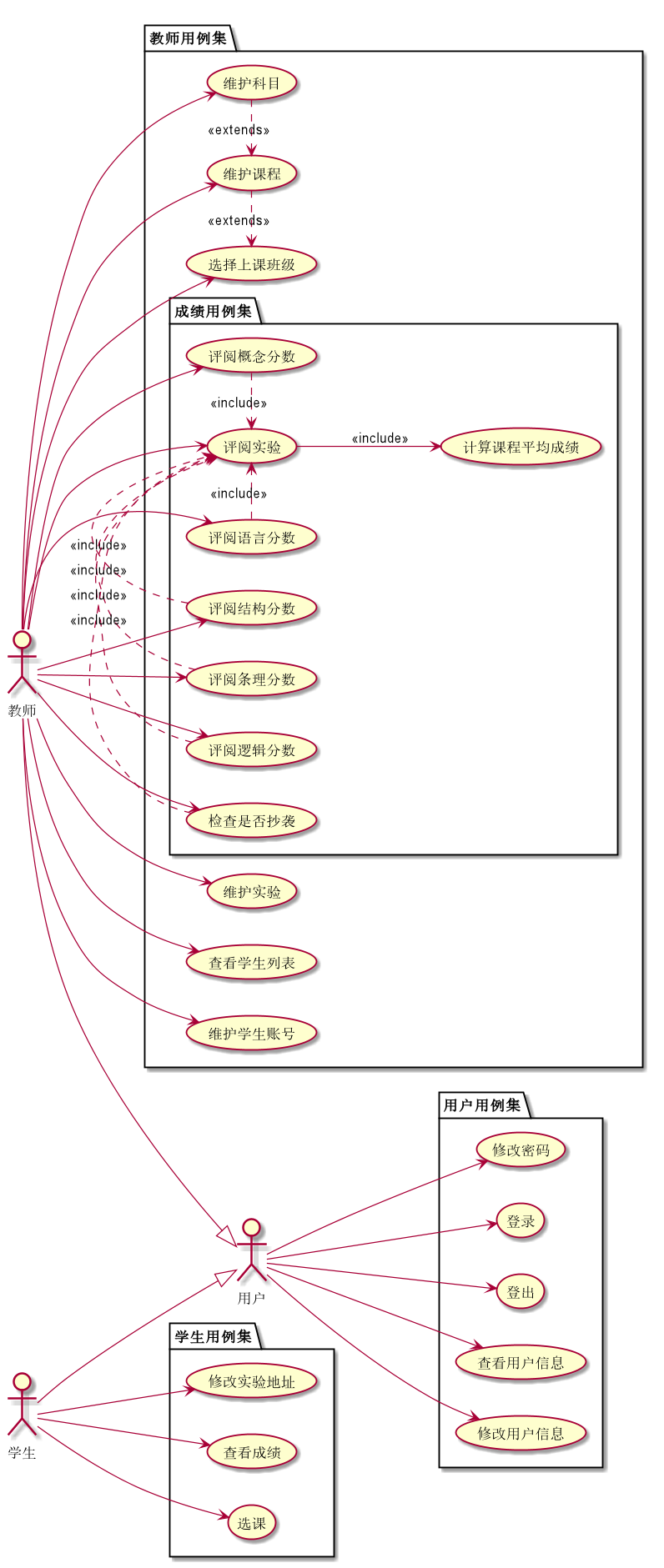
### **4.9 评分表**

数据库建表 table\_score.md

### **4.10 课程平均分表**

数据库建表 table\_average.md

## **5. 用例图设计**



用例图源码 flow\_usercase.puml

## **6. 用例设计**

### **6.1 登录用例**

用例规约表设计 table\_login.md  
用例顺序图设计 sequence\_login.png  
用例顺序图源码 sequence\_login.puml  
用例接口设计 api\_login.md  
用例界面设计 /user/login.html

### **6.2 登出用例**

用例规约表设计 table\_logout.md  
用例顺序图设计 sequence\_logout.png  
用例顺序图源码 sequence\_logout.puml  
用例接口设计 api\_logout.md  
用例界面设计 /user/logout.html

### **6.3 查看用户信息用例**

用例规约表设计 table\_user\_view.md  
用例顺序图设计 sequence\_user\_view.png  
用例顺序图源码 sequence\_user\_view.puml  
用例接口设计 api\_getUser.md  
用例界面设计 /user/get.html

### **6.4 修改用户信息用例**

用例规约表设计 table\_user\_update.md  
用例顺序图设计 sequence\_user\_update.png  
用例顺序图源码 sequence\_user\_update.puml  
用例接口设计 api\_setUser.md  
用例界面设计 /user/set.html

### **6.5 修改密码用例**

用例规约表设计 table\_password\_update.md  
用例顺序图设计 sequence\_password\_update.png  
用例顺序图源码 sequence\_password\_update.puml  
用例接口设计 api\_setPassword.md  
用例界面设计 /user/password/set.html

### **6.6 维护课程用例**

用例规约表设计 table\_lesson\_maintain.md  
用例顺序图设计 sequence\_lesson\_maintain.png  
用例顺序图源码 sequence\_lesson\_maintain.puml  
用例接口设计 api\_setLesson.md  
用例界面设计 /lesson/set.html

### **6.7 选择上课班级用例**

用例规约表设计 table\_class\_choose.md  
用例顺序图设计 sequence\_class\_choose.png  
用例顺序图源码 sequence\_class\_choose.puml  
用例接口设计 api\_chooseClass.md  
用例界面设计 /lesson/class/choose.html

### **6.8 维护实验用例**

用例规约表设计 table\_test\_maintain.md  
用例顺序图设计 sequence\_test\_maintain.png  
用例顺序图源码 sequence\_test\_maintain.puml  
用例接口设计 api\_setTest.md  
用例界面设计 /test/set.html

### **6.9 评阅分数用例**

用例规约表设计 table\_score\_concept.md  
用例顺序图设计 sequence\_score\_concept.png  
用例顺序图源码 sequence\_score\_concept.puml  
用例接口设计 api\_setConceptScore.md  
用例界面设计 /test/score/concept/set.html

### **6.10 评阅语言分数用例**

用例规约表设计 table\_score\_language.md  
用例顺序图设计 sequence\_score\_language.png  
用例顺序图源码 sequence\_score\_language.puml  
用例接口设计 api\_setLanguageScore.md  
用例界面设计 /test/score/language/set.html

### **6.11 评阅结构分数用例**

用例规约表设计 table\_score\_structure.md  
用例顺序图设计 sequence\_score\_structure.png  
用例顺序图源码 sequence\_score\_structure.puml  
用例接口设计 api\_setStructureScore.md  
用例界面设计 /test/score/srtructure/set.html

### **6.12 评阅条理分数用例**

用例规约表设计 table\_score\_method.md  
用例顺序图设计 sequence\_score\_method.png  
用例顺序图源码 sequence\_score\_method.puml  
用例接口设计 api\_setMethodScore.md  
用例界面设计 /test/score/method/set.html

### **6.13 评阅逻辑分数用例**

用例规约表设计 table\_score\_logic.md  
用例顺序图设计 sequence\_score\_logic.png  
用例顺序图源码 sequence\_score\_logic.puml  
用例接口设计 api\_setLogicScore.md  
用例界面设计 /test/score/logic/set.html

### **6.14 评阅实验用例**

用例规约表设计 table\_test\_mark.md  
用例顺序图设计 sequence\_test\_mark.png  
用例顺序图源码 sequence\_test\_mark.puml  
用例接口设计 api\_setTestScore.md  
用例界面设计 /test/score/set.html

### **6.15 计算课程平均成绩用例**

用例规约表设计 table\_lesson\_score.md  
用例顺序图设计 sequence\_lesson\_score.png  
用例顺序图源码 sequence\_lesson\_score.puml  
用例接口设计 api\_setLessonScore.md  
用例界面设计 /lesson/score/set.html

### **6.16 选课用例**

用例规约表设计 table\_lesson\_choose.md  
用例顺序图设计 sequence\_lesson\_choose.png  
用例顺序图源码 sequence\_lesson\_choose.puml  
用例接口设计 api\_chooseLesson.md  
用例界面设计 /lesson/choose.html

### **6.17 查看成绩用例**

用例规约表设计 table\_student\_score.md  
用例顺序图设计 sequence\_student\_score.png  
用例顺序图源码 sequence\_student\_score.puml  
用例接口设计 api\_getStudentScore.md  
用例界面设计 /student/score/get.html

# **数据库设计**

## **table\_user.md（用户表）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| user\_id | int | 主键 | 否 |  | 自动递增，无符号 | 用户的id，为user表的主键。 |
| user\_name | varchar(50) |  | 否 |  |  | 用户的真实姓名。 |
| github\_name | varchar(50) |  | 是 | null |  | 用户的github用户名。 |
| update\_time | datetime |  | 是 | null |  | 用户的github用户名的修改时间。 |
| password | varchar(50) |  | 是 | $user\_id |  | 用户的密码，默认值为用户的id。 |
| disable | tinyint |  | 是 | 0 |  | 用户是否被禁用，0为没有被禁用，其他值为被禁用。 |

数据库设计

## **table\_teacher.md（教师表）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| teacher\_id | varchar(50) | 主键 | 否 |  |  | 教师的教职工号，为teacher表的主键。 |
| user\_id | int | 外键 | 是 | null | $user.user\_id | 用户的id，为user表的外键。 |
| department | varchar(255) |  | 是 | null |  | 老师所属的部门。 |

# **数据库设计**

## **table\_student.md（学生表）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| student\_id | varchar(50) | 主键 | 否 |  |  | 学生的学号，为student表的主键。 |
| user\_id | int | 外键 | 是 | null | $user.user\_id | 用户的id，为user表的外键。 |
| class\_id | int | 外键 | 是 | null | $class.class\_id | 班级的id，为class表的外键。 |
| student\_average | int | 外键 | 是 | 0 | 无符号, student\_average <= 100 | 学生所有课程平均分。 |

# **数据库设计**

## **table\_class.md（班级表）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| class\_id | int | 主键 | 否 |  | 自动递增，无符号 | 班级的id，为class表的主键。 |
| grade | int |  | 否 |  | 无符号，grade <= system.time.year | 班级所在的年级。 |
| major | varchar(50) |  | 否 |  |  | 班级所在的专业。 |
| number | int |  | 否 |  |  | 班级的顺序，即几班。 |

# **数据库设计**

## **table\_subject.md（科目表）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| subject\_id | int | 主键 | 否 |  | 自动递增，无符号 | 科目的id，为subject表的主键。 |
| subject\_name | varchar(255) |  | 否 |  |  | 科目的名字。 |

# **数据库设计**

## **table\_lesson.md（课程表）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| lesson\_id | int | 主键 | 否 |  | 自动递增，无符号 | 课程的id，为lesson表的主键。 |
| subject\_id | int | 外键 | 否 |  | $subject.subject\_id | 科目的id，为subject表的外键。 |
| teacher\_id | varchar(50) | 外键 | 否 |  | $teacher.teacher\_id | 教师的教职工号，为teacher表的外键。 |
| class\_id | int | 外键 | 是 | null | $class.class\_id | 如果该课程是教师选班级，则为班级的id，为class表的外键，否则为空，表示由学生选课程。 |
| choose | tinyint |  | 是 | 0 |  | 如果该课程是学生选课程，则为1，表示学生自主选课，否则为0，表示由教师选班级。 |
| year | int |  | 否 |  |  | 开课的年份。 |
| semester | char |  | 否 |  | “上” or “下” | 开课的学期，仅能为“上”或“下”，“上”指上半学期，“下”指下半学期。 |
| set\_time | datetime |  | 是 | null |  | 课程的设置时间。 |

# **数据库设计**

## **table\_choose.md（选课表）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| choose\_id | int | 主键 | 否 |  | 自动递增，无符号 | 选课的id，为choose表的主键。 |
| lesson\_id | int | 外键 | 否 |  | $lesson.lesson\_id | 课程的id，为lesson表的外键。 |
| student\_id | varchar(50) | 外键 | 否 |  | $student.student\_id | 学生的学号，为student表的外键。 |
| choose\_time | datetime |  | 是 | null |  | 学生选课的时间。 |

# **数据库设计**

## **table\_test.md（实验表）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| test\_id | int | 主键 | 否 |  | 自动递增，无符号 | 实验的id，为test表的主键。 |
| lesson\_id | int | 外键 | 否 |  |  | 课程的id，为lesson表的外键。 |
| test\_name | vachar(255) |  | 否 |  |  | 实验的名称。 |
| test\_time | datetime |  | 是 | null | test\_time >= $lesson.year | 实验的开始时间。 |
| set\_time | datetime |  | 是 | null |  | 实验的设置时间。 |

# **数据库设计**

## **table\_score.md（评分表）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| score\_id | int | 主键 | 否 |  | 自动递增，无符号 | 成绩的id，为score表的主键。 |
| test\_id | int | 外键 | 否 |  | $test.test\_id | 成绩所属实验的id，为test表的外键。 |
| student\_id | varchar(50) | 外键 | 否 |  | $student.student\_id | 学生的学号，为student表的外键。 |
| github | varchar(255) |  | 是 | null |  | 学生该实验的github网址。 |
| concept | int |  | 是 | 0 | 无符号，concept <= 20 | 学生该实验的成绩--概念正确，20分制。 |
| language | int |  | 是 | 0 | 无符号，language <= 20 | 学生该实验的成绩--语言准确，20分制。 |
| structure | int |  | 是 | 0 | 无符号，structure <= 20 | 学生该实验的成绩--结构严谨，20分制。 |
| method | int |  | 是 | 0 | 无符号，method <= 20 | 学生该实验的成绩--条理清楚，20分制。 |
| logic | int |  | 是 | 0 | 无符号，logic <= 20 | 学生该实验的成绩--逻辑合理，20分制。 |
| plagiarize | tinyint |  | 是 | 0 |  | 学生该实验是否抄袭，0为没有抄袭，1为确认抄袭，若出现抄袭，该成绩直接0分。 |
| test\_score | int |  | 是 | 0 | 无符号，test\_score <= 100 | 学生该实验的总成绩。 |
| comment | varchar(255) |  | 是 | null |  | 教师对学生的评语。 |
| mark\_time | datetime |  | 是 | null |  | 老师批阅实验的时间。 |

# **数据库设计**

## **table\_average.md（课程平均分表）**

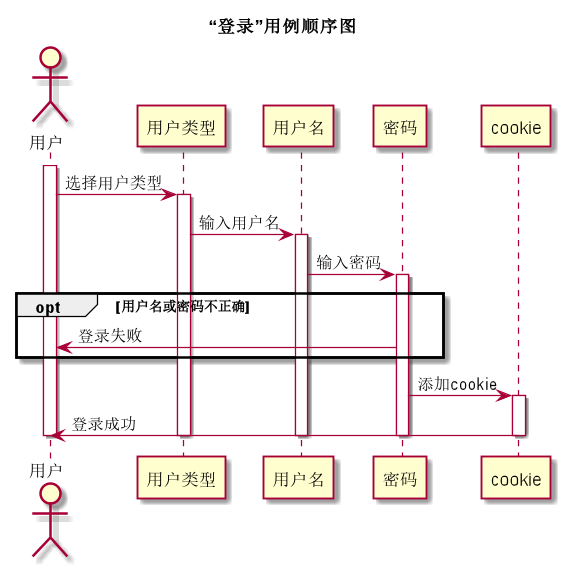
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 约束 | 说明 |
| average\_id | int | 主键 | 否 |  | 自动递增，无符号 | 平均分的id，为average表的主键。 |
| lesson\_id | int | 外键 | 否 |  | $lesson.lesson\_id | 平均分所属课程的id，为lesson表的外键。 |
| student\_id | varchar(50) | 外键 | 否 |  | $student.student\_id | 学生的学号，为student表的外键。 |
| lesson\_score | int |  | 否 |  | 无符号，lesson\_score <= 100 | 学生一个课程的平均成绩。 |
| count\_time | datetime |  | 是 | null |  | 课程平均分的计算时间。 |

# **用例规约表设计**

## **table\_login.md（登录用例规约表）**

“登录”用例规约

|  |  |
| --- | --- |
| 用例名称 | 登录 |
| 参与者 | 访客 |
| 前置条件 | 访客进入实验管理平台首页 |
| 后置条件 | 系统跳转至用户个人主页 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 访客进入登录界面； 2. 访客选择用户类型； 3. 访客输入用户名； 4. 访客输入密码； 5. 访客点击登录按钮； | 6. 系统判断用户名、密码、用户类型正确； 7. 系统在客户端以cookie形式存储登录用户信息； 8. 系统提示访客登录成功 |
| 备选事件流 | |
| 2a. 访客未选择用户类型        1. 系统提示选择用户类型，转第2步； 3a. 访客未输入用户名        1. 系统提示输入用户名，转第3步； 4a. 访客未输入用户名        1. 系统提示输入密码，转第4步； 5a. 访客选择返回        1. 系统跳转至实验管理平台主页面，转第1步； 6a. 系统验证登录时出现系统故障        1. 系统提示登录失败，转第5步； 6b. 系统验证登录失败        1. 系统提示用户名或密码错误，转第4步； 7a. 系统储存cookie时出现系统故障        1. 系统提示存储cookie失败，转第7步。 | |
| 业务规则 | |
| 1. 访客一次只能登录一个账号。 | |



# **接口设计**

## **api\_login.md（登录接口）**

* 权限：访客。
* 功能：用于用户登录
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/user/login/.set](http://phf449540929.github.io/is_analysis/v1/api/user/login/.set)
* 请求方法：GET
* 请求实例：

{

"access\_token": $access\_token,

"method": "GET",

"user\_id":"201710414305",

"password":"1",

"type":"student"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“GET”。 |
| user\_id | 是 | 学生的学号或者老师的教职工号，由type决定，对应表user.user\_id的值。 |
| password | 是 | 用户的密码，不是原文，是加密后的字符串，对应表user.password的值。 |
| type | 是 | 用户类型，学生或者老师。 |

* 返回实例：

{

"information": "登录成功",

"status": true,

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

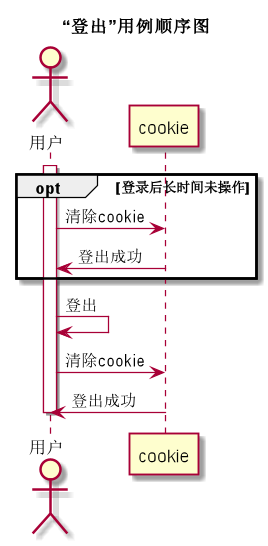


# **用例规约表设计**

## **table\_logout.md（登出用例规约表）**

“登出”用例规约

|  |  |
| --- | --- |
| 用例名称 | 登出 |
| 参与者 | 学生/教师 |
| 前置条件 | 用户已登录 |
| 后置条件 | 系统跳转至实验管理平台首页 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 用户点击登出按钮；  3. 用户点击确认登出按钮； | 2. 系统提示是否确认登出；  4. 系统清除用户的登录信息； 5. 系统清除客户端登录信息cookie; 6. 系统提示用户登出成功。 |
| 备选事件流 | |
| 1a. 用户登录之后，长时间不操作界面        1. 系统默认登出，转第4步； 3a. 用户选择取消登出        1. 系统返回用户个人主页，转第1步； 4a. 系统清楚用户登录信息时出现系统故障        1. 系统提示登出失败，转第1步； 5a. 系统清楚cookie信息时出现系统故障        1. 系统提示清楚cookie信息失败，转第1步。 | |
| 业务规则 | |
| 1. 用户对于已经登录的账户只能登出一次。 | |



# **接口设计**

## **api\_logout.md（登出接口）**

* 权限：学生/教师已经登录，即接口参数user\_id必须等于登录用户的user\_id。
* 功能：用于用户登出
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/user/logout/.set](http://phf449540929.github.io/is_analysis/v1/api/user/logout/.set)
* 请求方法：GET
* 请求实例：

{

"access\_token": $access\_token,

"method": "GET",

"user\_id":"201710414305"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“GET”。 |
| user\_id | 是 | 学生的学号或者老师的教职工号，由type决定，对应表user.user\_id的值。 |

* 返回实例：

{

"information": "登出成功",

"status": true,

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

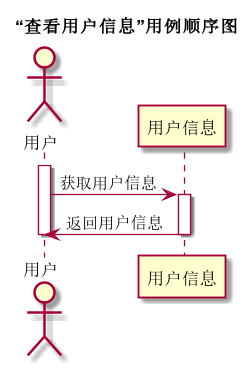


# **用例规约表设计**

## **table\_user\_view.md（查看用户信息用例规约表）**

“查看用户信息”用例规约

|  |  |
| --- | --- |
| 用例名称 | 查看用户信息 |
| 参与者 | 学生/教师 |
| 前置条件 | 用户已登录 |
| 后置条件 | 系统跳转至用户个人信息页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 用户点击查看用户信息按钮； | 2. 系统验证用户登录状态是否有效； 3. 系统获取用户信息； 4. 系统返回用户信息。 |
| 备选事件流 | |
| 2a. 系统验证用户登录状态已失效        1. 系统提示验证用户登录状态失败，转第1步； 3a. 系统获取用户信息时出现系统故障        1. 系统提示获取用户信息失败，转第1步。 | |
| 业务规则 | |
| 1. 用户只能查询用户自己的信息。 | |



# **接口设计**

## **api\_getUser.md（查看用户信息接口）**

* 权限：学生/教师：查看自己的信息，必须先登录，不能查看其他用户的信息，即接口参数user\_id必须等于登录用户的user\_id。
* 功能：用于用户查看用户信息
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/user/.get](http://phf449540929.github.io/is_analysis/v1/api/user/.get)
* 请求方法：GET
* 请求实例：

{

"access\_token": $access\_token,

"method": "GET",

"user\_id": 1

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“GET”。 |
| user\_id | 是 | 用户的id，对应表user.user\_id的值。 |

* 返回实例：

{

"information": "查看用户信息成功",

"status": true,

"user\_id": 1,

"type": "学生"

"student\_id": "201710414305",

"teacher\_id": null,

"user\_name": "胡宏坤",

"class\_id": 1,

"github\_name": "NamelessSoldier",

"password": "AF#W@#AAAASDF",

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| user\_id | 是 | 用户的id，对应表user.user\_id的值。 |
| type | 是 | 用户的类型，用户为学生则为“学生”，用户为教师则为“教师”。 |
| student\_id | 否 | 学生的学号，若用户类型为“教师”则为空，对应表student.student\_id的值。 |
| teacher\_id | 否 | 教师的教职工号，若用户类型为“学生”则为空，对应表teacher.teacher\_id的值。 |
| user\_name | 是 | 用户的真实姓名，对应表user.user\_name的值。 |
| class\_id | 是 | 班级的id，对应表class.class\_id的值。 |
| github\_name | 否 | 用户的github用户名，对应表user.github的值。 |
| password | 是 | 用户的密码，不能为空，必须经过加密，不能是密码的原文，对应表user.password的值。 |
| code | 是 | 返回码。 |

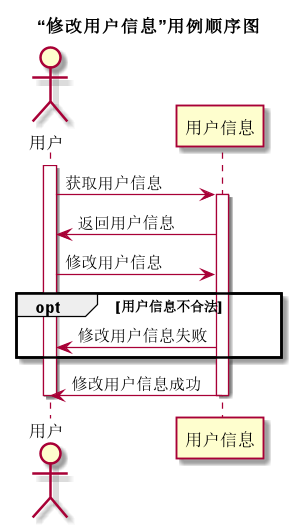


# **用例规约表设计**

## **table\_user\_update.md（修改用户信息用例规约表）**

“修改用户信息”用例规约

|  |  |
| --- | --- |
| 用例名称 | 修改用户信息 |
| 参与者 | 学生/教师 |
| 前置条件 | 用户已登录，并进入个人信息页面 |
| 后置条件 | 系统跳转至用户个人信息页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 用户点击修改用户信息按钮；    5. 用户填写需要修改的信息； 6. 用户点击提交按钮； | 2. 系统验证用户登录状态是否有效； 3. 系统获取用户信息； 4. 系统跳转至修改用户信息页面；   7. 系统检测用户提交的修改信息是否合法； 8. 系统存储修改后的用户信息； 9. 系统提示用户信息修改成功。 |
| 备选事件流 | |
| 2a. 系统验证用户登录状态已失效        1. 系统提示验证用户登录状态失败，转第1步； 3a. 系统获取用户信息时出现系统故障        1. 系统提示获取用户信息失败，转第1步； 6a. 用户选择返回，放弃修改        1. 系统跳转至用户个人信息主页，转第1步； 7a. 系统检测到用户提交的修改信息非法        1. 系统提示用户提交信息非法，转第5步； 8a. 系统存储信息时出现系统故障        1. 系统提示修改用户信息失败，转第6步。 | |
| 业务规则 | |
| 1. 用户只能修改用户自己的信息； 2. 用户提交的修改信息必须合法。 | |



# **接口设计**

## **api\_setUser.md（修改用户信息接口）**

* 权限：学生/老师：修改自己的密码，必须先登录，即接口参数user\_id必须等于登录用户的user\_id。
* 功能：用于用户修改用户信息
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/user/.set](http://phf449540929.github.io/is_analysis/v1/api/user/.set)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"user\_id": "201710414305",

"user\_name": "胡宏坤",

"github\_name": "NamelessSoldier",

"password": "AF#W@#AAAASDF"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| user\_id | 是 | 用户的id，对应表user.user\_id的值。 |
| user\_name | 是 | 用户的真实姓名，对应表user.user\_name的值。 |
| github\_name | 否 | 用户的github用户名，对应表user.github的值。 |
| password | 是 | 用户的密码，不能为空，必须经过加密，不能是密码的原文，对应表user.password的值。 |

* 返回实例：

{

"information": "修改用户信息成功",

"status": true,

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

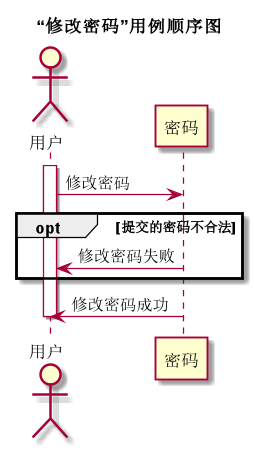


# **用例规约表设计**

## **table\_password\_update.md（修改密码用例规约表）**

“修改密码信息”用例规约

|  |  |
| --- | --- |
| 用例名称 | 修改密码 |
| 参与者 | 学生/教师 |
| 前置条件 | 用户已登录 |
| 后置条件 | 系统跳转至实验管理平台首页 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 用户点击修改密码按钮；   4. 用户填写修改的密码； 5. 用户点击提交按钮； | 2. 系统验证用户登录状态是否有效； 3. 系统跳转至修改密码页面；   6. 系统检测用户提交的密码是否合法； 7. 系统存储修改后的密码； 8. 系统提示密码修改成功； 9. 系统强制登出用户。 |
| 备选事件流 | |
| 2a. 系统验证用户登录状态已失效        1. 系统提示验证用户登录状态失败，转第1步； 5a. 用户选择返回，放弃修改        1. 系统跳转至用户个人信息主页，转第1步； 6a. 系统检测到用户提交的密码非法        1. 系统提示用户提交密码非法，转第4步； 7a. 系统存储密码时出现系统故障        1. 系统提示修改密码失败，转第5步。 | |
| 业务规则 | |
| 1. 用户只能修改用户自己的密码； 2. 用户提交的密码必须合法。 | |



# **接口设计**

## **api\_setPassword.md（修改密码接口）**

* 权限： 学生/老师：修改自己的密码，必须先登录，即接口参数user\_id必须等于登录用户的user\_id。
* 功能：用于用户修改密码
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/user/password/.set](http://phf449540929.github.io/is_analysis/v1/api/user/password/.set)
* 请求方法：UPDATE
* 请求实例：

{

"access\_token": $access\_token,

"method": "UPDATE",

"user\_id":"201710414305",

"password": "AF#W@#AAAASDF"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“UPDATE”。 |
| user\_id | 是 | 学生的学号或者老师的教职工号，对应表user.user\_id的值。 |
| password | 是 | 用户的密码，不能为空，必须经过加密，不能是密码的原文，对应表user.password的值。 |

* 返回实例：

{

"information": "修改密码成功",

"status": true,

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

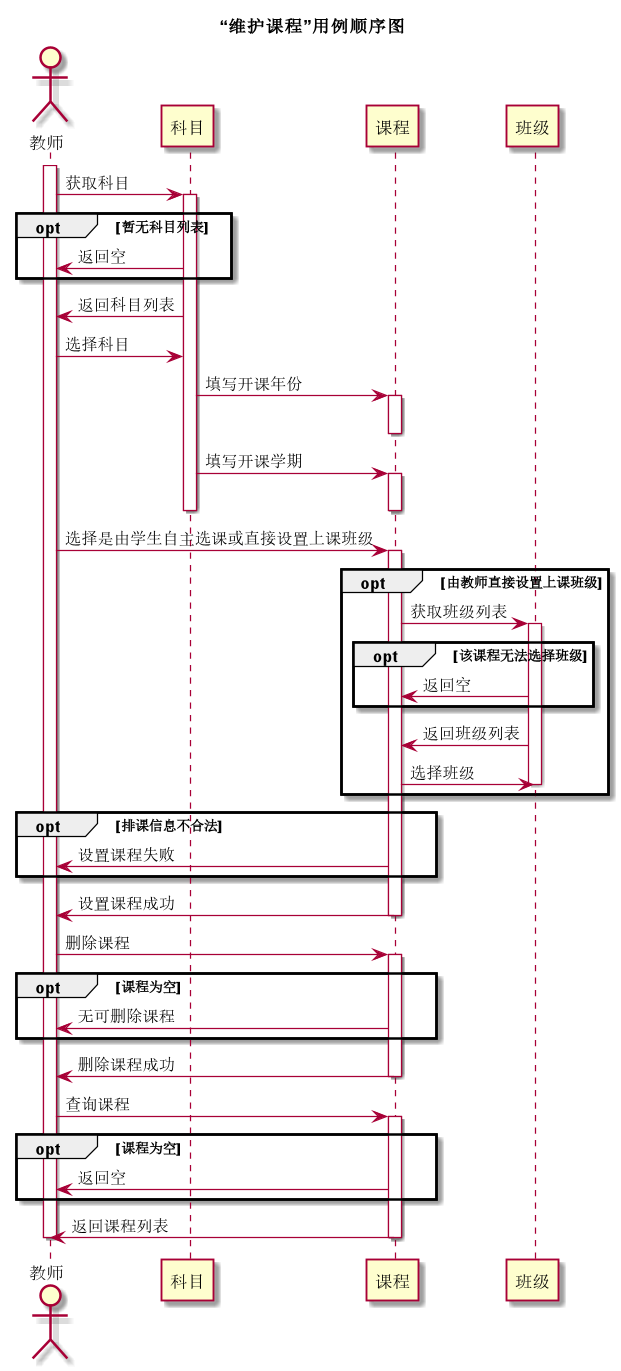


# **用例规约表设计**

## **table\_lesson\_maintain.md（维护课程用例规约表）**

“维护课程”用例规约

|  |  |
| --- | --- |
| 用例名称 | 维护课程 |
| 参与者 | 教师 |
| 前置条件 | 教师已登录 |
| 后置条件 | 系统跳转至课程实验页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 教师点击排课按钮；   4. 教师选择科目； 5. 教师填写课程开课年份； 6. 教师填写课程开课学期； 7. 教师选择课程由学生自主选课或直接选择上课班级； 8. 教师点击提交按钮； | 2. 系统验证教师登录状态是否有效； 3. 系统跳转至排课页面；      9. 系统检测提交的课程信息是否合法； 10.系统存储课程信息； 11.系统提示排课成功。 |
| 备选事件流 | |
| 2a. 系统验证教师登录状态已失效        1. 系统提示验证教师登录状态失败，转第1步； 5a. 教师未填写开课年份        1. 系统提示填写开课年份，转第5步； 6a. 教师未填写开课学期        1. 系统提示填写开课学期，转第6步； 7a. 教师选择直接选择班级        1. 系统提示教师选择上课班级；        2. 教师选择上课班级，转第8步； 8a. 教师选择返回，放弃排课        1. 系统跳转至实验管理平台首页，转第1步； 9a. 系统检测到提交的课程信息非法        1. 系统提示提交的课程信息非法，转第4步； 9b. 系统检测到开课年份在当前时间之前        1. 系统提示填写的开课年份错误，转第5步； 10a.系统存储排课信息时出现系统故障        1. 系统提示排课失败，转第8步。 | |
| 业务规则 | |
| 1. 教师可以设置课程为学生选课或直接选择班级； 2. 教师一次只能排一门课； 3. 教师安排的课程信息必须合法； 4. 教师设置的开课年份必须在当前时间之后； | |



# **接口设计**

## **api\_setLesson.md（维护课程级接口）**

* 权限：教师：只能维护已存在学科的课程，即接口参数subject\_id必须存在以其为主键的subject表，只能开设自己为任课老师的课程，即接口参数teacher\_id必须等于登录教师的teacher\_id。
* 功能：用于教师维护课程
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/lesson/.set](http://phf449540929.github.io/is_analysis/v1/api/lesson/.set)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"subject\_id": 1,

"teacher\_id": "201710414305",

"choose": true,

"year": 2020,

"semester": "第二学期"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| subject\_id | 是 | 科目的id，对应表subject.subject\_id的值。 |
| teacher\_id | 是 | 教师的教职工号，对应表teacher.teahcer\_id的值。 |
| choose | 是 | 是否由学生自主选课，“是”表示由学生自主选课，“否”表示由教师直接选择上课班级，对应表lesson.choose的值。 |
| year | 是 | 开课年份，对应表lesson.year的值。 |
| semester | 是 | 开课学期，对应表lesson.semester的值。 |

* 返回实例：

{

"information": "维护课程成功",

"status": true,

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

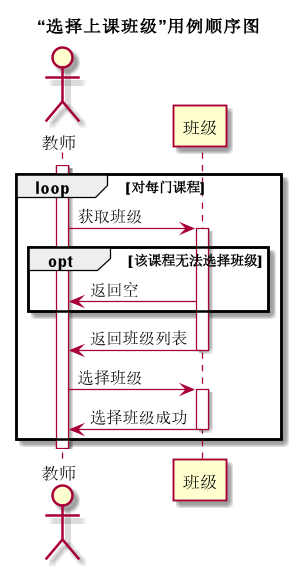


# **用例规约表设计**

## **table\_class\_choose.md（选择上课班级用例规约表）**

“选择上课班级”用例规约

|  |  |
| --- | --- |
| 用例名称 | 选择上课班级 |
| 参与者 | 教师 |
| 前置条件 | 教师已登录，并进入排课流程，选择了直接选择上课班级 |
| 后置条件 | 系统跳转至课程实验页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 教师对课程选择直接选择上课班级；     6. 教师选择上课班级； 7. 教师点击提交按钮； | 2. 系统验证教师登录状态是否有效； 3. 系统检索可以被选择的班级； 4. 系统返回检索到的班级； 5. 系统跳转至选择上课班级页面；   8. 系统存储课程的上课班级信息； 9. 系统提示选择上课班级成功。 |
| 备选事件流 | |
| 2a. 系统验证教师登录状态已失效        1. 系统提示验证教师登录状态失败，转第1步； 3a. 系统检索到没有可以被选择的班级        1. 系统提示无班级可供选择，转第1步； 3b. 系统在检索班级时出现系统故障        1. 系统提示检索班级失败，转第1步； 7a. 教师选择返回，放弃选择上课班级        1. 系统跳转至排课页面，转第1步； 8a. 系统存储上课班级信息时出现系统故障        1. 系统提示选择上课班级失败，转第6步。 | |
| 业务规则 | |
| 1. 教师可以为一门课程选择多个班级； 2. 教师只能选择可以被选择的班级。 | |



# **接口设计**

## **api\_chooseClass.md（选择上课班级接口）**

* 权限：教师：只能为自己开设的课程选择上课班级，即接口参数lesson\_id所对应lesson表中的lesson.teacher\_id必须等于登录教师的teacher\_id。
* 功能：用于选择上课班级
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/lesson/class/.choose](http://phf449540929.github.io/is_analysis/v1/api/lesson/class/.choose)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"lesson\_id": 1,

"class\_id": 11

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| lesson\_id | 是 | 课程的id，对应表lesson.lesson\_id的值。 |
| class\_id | 是 | 班级的id，对应表class.class\_id的值。 |

* 返回实例：

{

"information": "设置班级成功",

"status": true,

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

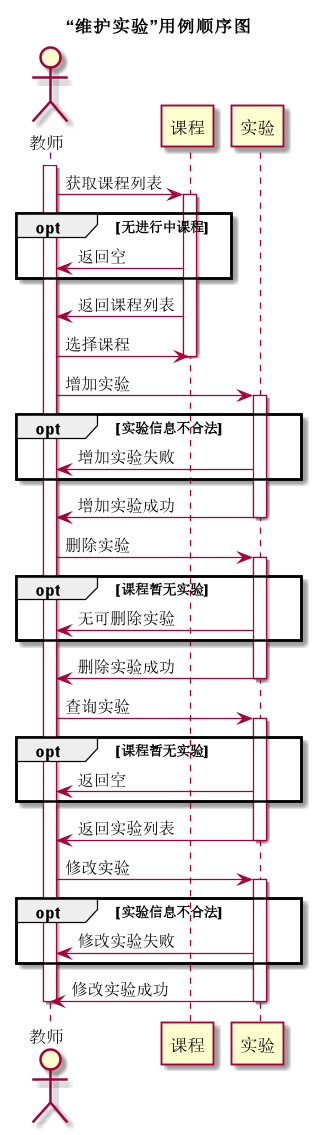


# **用例规约表设计**

## **table\_test\_maintain.md（维护实验用例规约表）**

“维护实验”用例规约

|  |  |
| --- | --- |
| 用例名称 | 维护实验 |
| 参与者 | 教师 |
| 前置条件 | 教师已登录，并进入课程主页 |
| 后置条件 | 系统跳转至课程实验页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 教师点击开设实验按钮；   4. 教师填写实验信息； 5. 教师填写实验日期； 6. 教师点击提交按钮； | 2. 系统验证教师登录状态是否有效； 3. 系统跳转至开设实验页面；    7. 系统检测实验日期是否在当前时间之后； 8. 系统检测实验信息是否合法； 9. 系统存储实验信息； 10.系统提示开设实验成功。 |
| 备选事件流 | |
| 2a. 系统验证教师登录状态已失效        1. 系统提示验证教师登录状态失败，转第1步； 5a. 教师选择返回，放弃开设实验        1. 系统跳转至课程实验页面，转第1步； 7a. 系统检测到实验日期在当前时间之后        1. 系统提示实验日期在当前时间之后，转第5步； 8a. 系统检测到实验信息不合法        1. 系统提示实验信息不合法，转第4步； 9a. 系统存储实验信息时出现系统故障        1. 系统提示开设实验失败，转第6步。 | |
| 业务规则 | |
| 1. 教师一次只能设置一门课程的一个实验； 2. 教师提交的实验信息必须合法； 3. 教师提交的实验日期必须在当前时间之后。 | |



# **接口设计**

## **api\_setTest.md（维护实验接口）**

* 权限：教师：只能维护自己开设课程的实验，即接口参数lesson\_id所对应lesson表中的lesson.teacher\_id必须等于登录教师的teacher\_id。
* 功能：用于教师维护实验
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/test/.set](http://phf449540929.github.io/is_analysis/v1/api/test/.set)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"lesson\_id": 6,

"test\_name": "基于GitHub的实验管理平台",

"test\_time": 2020-05-26 08:30:00

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| lesson\_id | 是 | 课程的id，对应表lesson.lesson\_id的值。 |
| test\_name | 是 | 实验的名字，对应表test.test\_name的值。 |
| test\_time | 是 | 实验的开始时间，对应表test.test\_time的值。 |

* 返回实例：

{

"information": "维护实验成功",

"status": true,

"code": 200

}

* 返回参数说明：

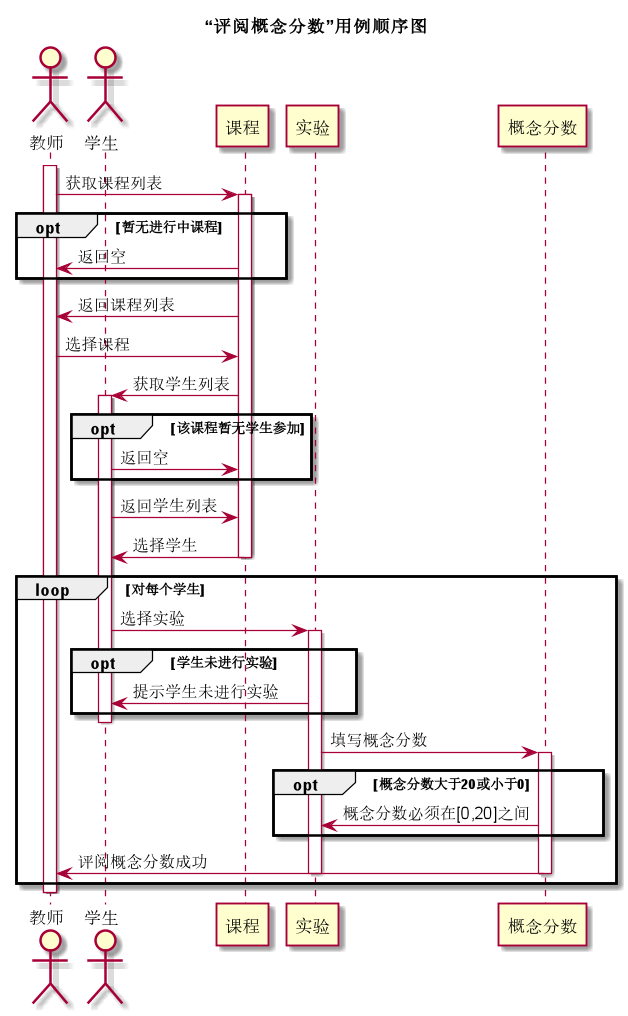
|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

# **用例规约表设计**

## **table\_score\_concept.md（评阅概念分数用例规约表）**

“评阅概念分数”用例规约

|  |  |
| --- | --- |
| 用例名称 | 评阅概念分数 |
| 参与者 | 教师 |
| 前置条件 | 教师已登录，并进入课程实验页面，获取到学生列表 |
| 后置条件 | 系统跳转至实验的学生列表页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 教师选择学生；   4. 教师填写实验的概念分数； 5. 教师点击提交按钮； | 2. 系统验证教师登录状态是否有效； 3. 系统跳转至实验评分页面；   6. 系统检测概念分数是否大于0； 7. 系统检测概念分数是否小于20； 8. 系统存储评阅的概念分数； 9. 系统提示评阅实验的概念分数成功。 |
| 备选事件流 | |
| 2a. 系统验证教师登录状态已失效        1. 系统提示验证教师登录状态失败，转第1步； 5a. 教师选择返回，放弃评阅概念分数        1. 系统跳转至实验的学生列表页面，转第1步； 6a. 系统检测到概念分数为负        1. 系统提示概念分数必须为正，转第4步； 7a. 系统检测到概念分数大于20        1. 系统提示概念分数必须小于20，转第4步； 8a. 系统存储实验的概念分数时出现系统故障        1. 系统提示评阅概念分数失败，转第5步。 | |
| 业务规则 | |
| 1. 教师一次只能评阅一个课程中一个学生的一个实验； 2. 教师填写的概念分数必须为正，且小于20； 3. 教师可以重复评阅一个实验的概念分数。 | |



# **接口设计**

## **api\_setConceptScore.md（评阅概念分数接口）**

* 权限：教师：只能评阅参加自己开设课程的学生的实验概念分数，即接口参数test\_id对应的test表中的test.lesson\_id所对应的lesson表中的lesson.teacher\_id必须等于登录教师的teacher\_id，接口参数student\_id必须等于test\_id对应的test表中的test.lesson\_id所对应的choose表中的choose.student\_id或所对应lesson.class\_id必须等于对应student表中的student.class\_id。
* 功能：用于教师评阅概念分数
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/test/score/concept/.set](http://phf449540929.github.io/is_analysis/v1/api/test/score/concept/.set)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"student\_id": "201710414305",

"test\_id": 1,

"concept": "19"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| student\_id | 是 | 学生的学号，对应表student.student\_id的值。 |
| test\_id | 是 | 实验的id，对应表test.test\_id的值。 |
| concept | 是 | 概念的分数，对应表score.concept的值。 |

* 返回实例：

{

"information": "评阅概念分数成功",

"status": true,

"code": 200

}

* 返回参数说明：

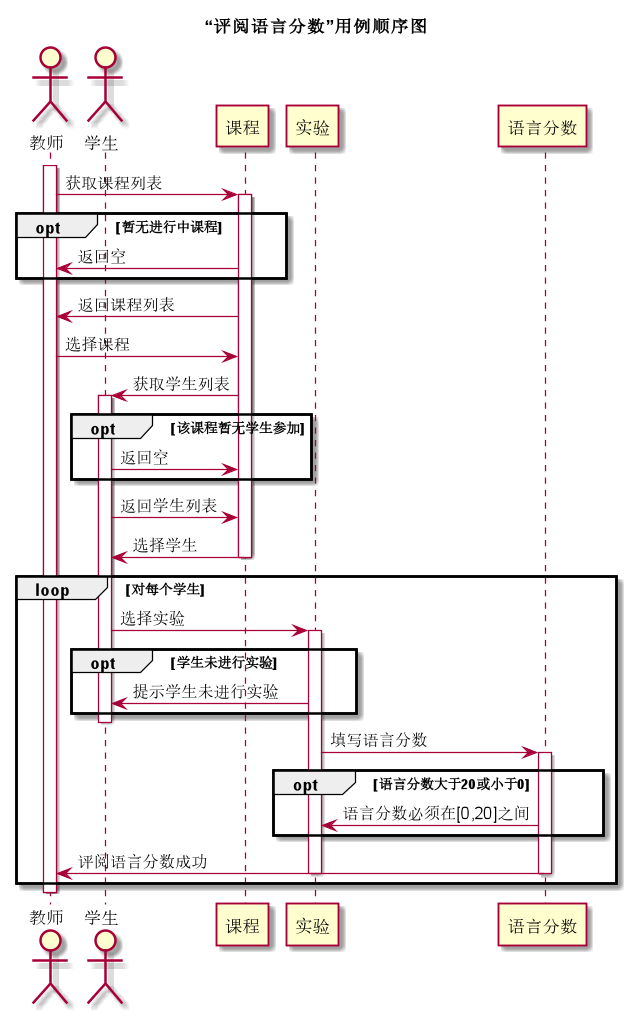
|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

# **用例规约表设计**

## **table\_score\_language.md（评阅语言分数用例规约表）**

“评阅语言分数”用例规约

|  |  |
| --- | --- |
| 用例名称 | 评阅语言分数 |
| 参与者 | 教师 |
| 前置条件 | 教师已登录，并进入课程实验页面，获取到学生列表 |
| 后置条件 | 系统跳转至实验的学生列表页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 教师选择学生；   4. 教师填写实验的语言分数； 5. 教师点击提交按钮； | 2. 系统验证教师登录状态是否有效； 3. 系统跳转至实验评分页面；   6. 系统检测语言分数是否大于0； 7. 系统检测语言分数是否小于20； 8. 系统存储评阅的语言分数； 9. 系统提示评阅实验的语言分数成功。 |
| 备选事件流 | |
| 2a. 系统验证教师登录状态已失效        1. 系统提示验证教师登录状态失败，转第1步； 5a. 教师选择返回，放弃评阅语言分数        1. 系统跳转至实验的学生列表页面，转第1步； 6a. 系统检测到语言分数为负        1. 系统提示语言分数必须为正，转第4步； 7a. 系统检测到语言分数大于20        1. 系统提示语言分数必须小于20，转第4步； 8a. 系统存储实验的语言分数时出现系统故障        1. 系统提示评阅语言分数失败，转第5步。 | |
| 业务规则 | |
| 1. 教师一次只能评阅一个课程中一个学生的一个实验； 2. 教师填写的语言分数必须为正，且小于20； 3. 教师可以重复评阅一个实验的语言分数。 | |



# **接口设计**

## **api\_setLanguageScore.md（评阅语言分数接口）**

* 权限：教师：只能评阅参加自己开设课程的学生的实验语言分数，即接口参数test\_id对应的test表中的test.lesson\_id所对应的lesson表中的lesson.teacher\_id必须等于登录教师的teacher\_id，接口参数student\_id必须等于test\_id对应的test表中的test.lesson\_id所对应的choose表中的choose.student\_id或所对应lesson.class\_id必须等于对应student表中的student.class\_id。
* 功能：用于教师评阅语言分数
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/test/score/language/.set](http://phf449540929.github.io/is_analysis/v1/api/test/score/language/.set)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"student\_id": "201710414305",

"test\_id": 1,

"language": "19"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| student\_id | 是 | 学生的学号，对应表student.student\_id的值。 |
| test\_id | 是 | 实验的id，对应表test.test\_id的值。 |
| language | 是 | 语言的分数，对应表score.language的值。 |

* 返回实例：

{

"information": "评阅语言分数成功",

"status": true,

"code": 200

}

* 返回参数说明：

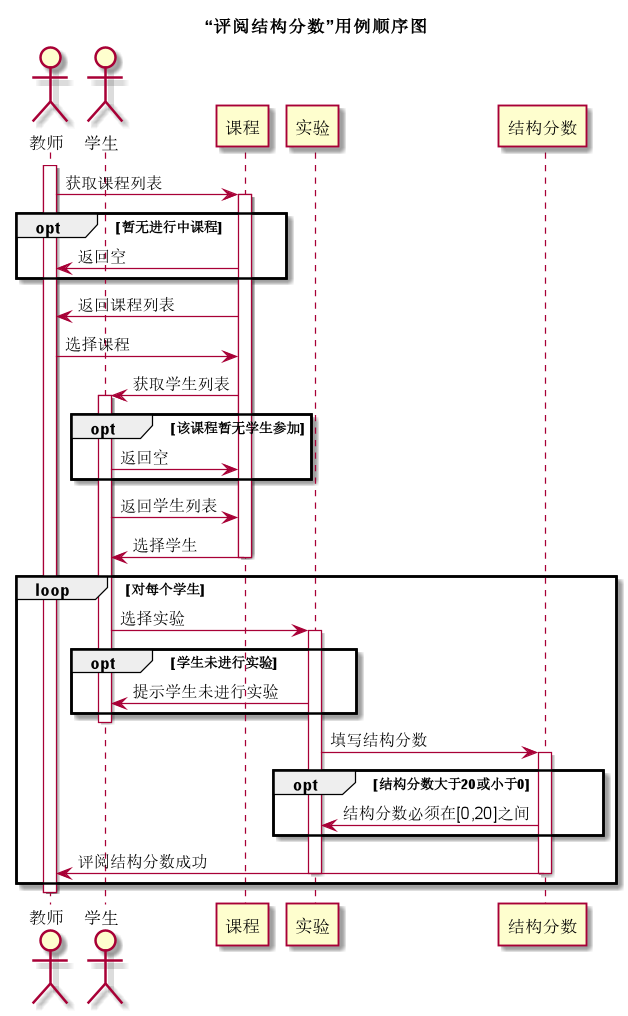
|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

# **用例规约表设计**

## **table\_score\_structure.md（评阅结构分数用例规约表）**

“评阅结构分数”用例规约

|  |  |
| --- | --- |
| 用例名称 | 评阅结构分数 |
| 参与者 | 教师 |
| 前置条件 | 教师已登录，并进入课程实验页面，获取到学生列表 |
| 后置条件 | 系统跳转至实验的学生列表页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 教师选择学生；   4. 教师填写实验的结构分数； 5. 教师点击提交按钮； | 2. 系统验证教师登录状态是否有效； 3. 系统跳转至实验评分页面；   6. 系统检测结构分数是否大于0； 7. 系统检测结构分数是否小于20； 8. 系统存储评阅的结构分数； 9. 系统提示评阅实验的结构分数成功。 |
| 备选事件流 | |
| 2a. 系统验证教师登录状态已失效        1. 系统提示验证教师登录状态失败，转第1步； 5a. 教师选择返回，放弃评阅结构分数        1. 系统跳转至实验的学生列表页面，转第1步； 6a. 系统检测到结构分数为负        1. 系统提示结构分数必须为正，转第4步； 7a. 系统检测到结构分数大于20        1. 系统提示结构分数必须小于20，转第4步； 8a. 系统存储实验的结构分数时出现系统故障        1. 系统提示评阅结构分数失败，转第5步。 | |
| 业务规则 | |
| 1. 教师一次只能评阅一个课程中一个学生的一个实验； 2. 教师填写的结构分数必须为正，且小于20； 3. 教师可以重复评阅一个实验的结构分数。 | |



# **接口设计**

## **api\_setStructureScore.md（评阅结构分数接口）**

* 权限：教师：只能评阅参加自己开设课程的学生的实验结构分数，即接口参数test\_id对应的test表中的test.lesson\_id所对应的lesson表中的lesson.teacher\_id必须等于登录教师的teacher\_id，接口参数student\_id必须等于test\_id对应的test表中的test.lesson\_id所对应的choose表中的choose.student\_id或所对应lesson.class\_id必须等于对应student表中的student.class\_id。
* 功能：用于教师评阅结构分数
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/test/score/structure/.set](http://phf449540929.github.io/is_analysis/v1/api/test/score/structure/.set)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"student\_id": "201710414305",

"test\_id": 1,

"structure": "19"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| student\_id | 是 | 学生的学号，对应表student.student\_id的值。 |
| test\_id | 是 | 实验的id，对应表test.test\_id的值。 |
| structure | 是 | 结构的分数，对应表score.structure的值。 |

* 返回实例：

{

"information": "评阅结构分数成功",

"status": true,

"code": 200

}

* 返回参数说明：

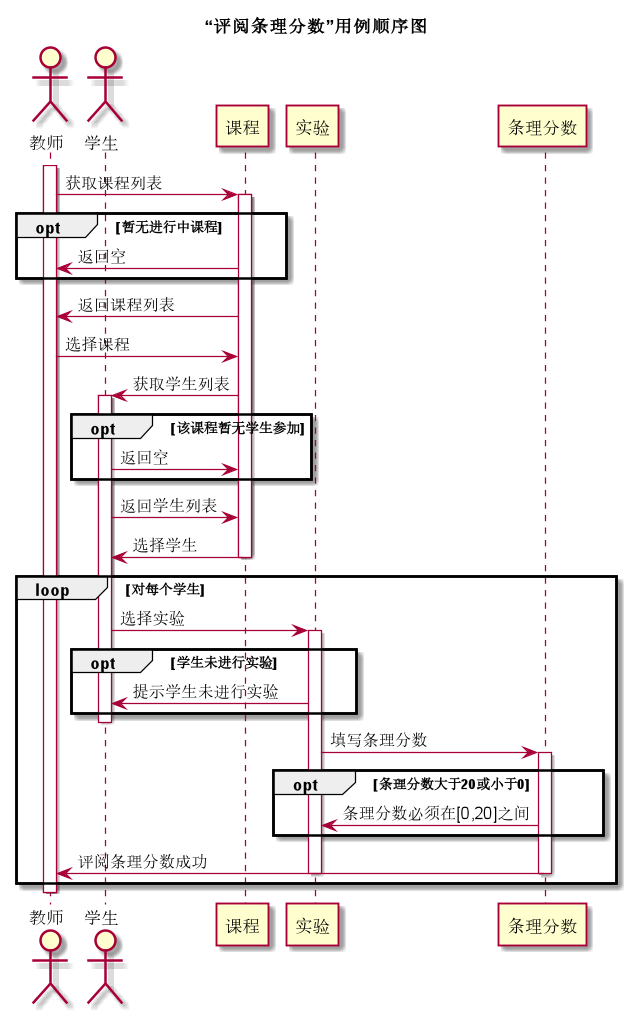
|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

# **用例规约表设计**

## **table\_score\_method.md（评阅条理分数用例规约表）**

“评阅条理分数”用例规约

|  |  |
| --- | --- |
| 用例名称 | 评阅条理分数 |
| 参与者 | 教师 |
| 前置条件 | 教师已登录，并进入课程实验页面，获取到学生列表 |
| 后置条件 | 系统跳转至实验的学生列表页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 教师选择学生；   4. 教师填写实验的条理分数； 5. 教师点击提交按钮； | 2. 系统验证教师登录状态是否有效； 3. 系统跳转至实验评分页面；   6. 系统检测条理分数是否大于0； 7. 系统检测条理分数是否小于20； 8. 系统存储评阅的条理分数； 9. 系统提示评阅实验的条理分数成功。 |
| 备选事件流 | |
| 2a. 系统验证教师登录状态已失效        1. 系统提示验证教师登录状态失败，转第1步； 5a. 教师选择返回，放弃评阅条理分数        1. 系统跳转至实验的学生列表页面，转第1步； 6a. 系统检测到条理分数为负        1. 系统提示条理分数必须为正，转第4步； 7a. 系统检测到条理分数大于20        1. 系统提示条理分数必须小于20，转第4步； 8a. 系统存储实验的条理分数时出现系统故障        1. 系统提示评阅条理分数失败，转第5步。 | |
| 业务规则 | |
| 1. 教师一次只能评阅一个课程中一个学生的一个实验； 2. 教师填写的条理分数必须为正，且小于20； 3. 教师可以重复评阅一个实验的条理分数。 | |



# **接口设计**

## **api\_setMethodScore.md（评阅条理分数接口）**

* 权限：教师：只能评阅参加自己开设课程的学生的实验条理分数，即接口参数test\_id对应的test表中的test.lesson\_id所对应的lesson表中的lesson.teacher\_id必须等于登录教师的teacher\_id，接口参数student\_id必须等于test\_id对应的test表中的test.lesson\_id所对应的choose表中的choose.student\_id或所对应lesson.class\_id必须等于对应student表中的student.class\_id。
* 功能：用于教师评阅条理分数
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/test/score/method/.set](http://phf449540929.github.io/is_analysis/v1/api/test/score/method/.set)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"student\_id": "201710414305",

"test\_id": 1,

"method\_score": "19"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| student\_id | 是 | 学生的学号，对应表student.student\_id的值。 |
| test\_id | 是 | 实验的id，对应表test.test\_id的值。 |
| method\_score | 是 | 条理的分数，对应表score.method的值。 |

* 返回实例：

{

"information": "评阅条理分数成功",

"status": true,

"code": 200

}

* 返回参数说明：

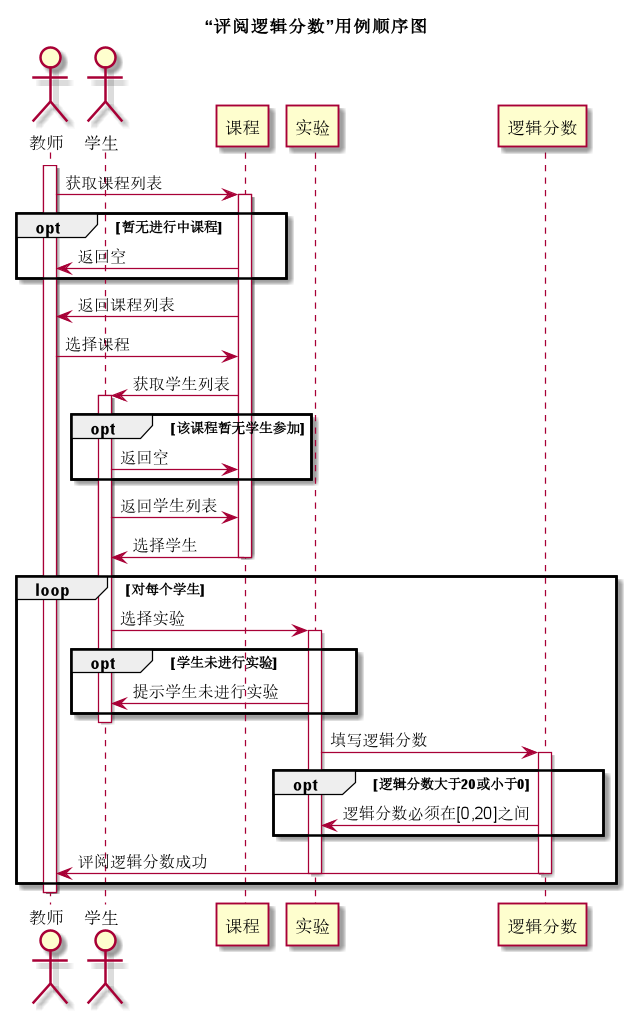
|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

# **用例规约表设计**

## **table\_score\_logic.md（评阅逻辑分数用例规约表）**

“评阅逻辑分数”用例规约

|  |  |
| --- | --- |
| 用例名称 | 评阅逻辑分数 |
| 参与者 | 教师 |
| 前置条件 | 教师已登录，并进入课程实验页面，获取到学生列表 |
| 后置条件 | 系统跳转至实验的学生列表页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 教师选择学生；   4. 教师填写实验的逻辑分数； 5. 教师点击提交按钮； | 2. 系统验证教师登录状态是否有效； 3. 系统跳转至实验评分页面；   6. 系统检测逻辑分数是否大于0； 7. 系统检测逻辑分数是否小于20； 8. 系统存储评阅的逻辑分数； 9. 系统提示评阅实验的逻辑分数成功。 |
| 备选事件流 | |
| 2a. 系统验证教师登录状态已失效        1. 系统提示验证教师登录状态失败，转第1步； 5a. 教师选择返回，放弃评阅逻辑分数        1. 系统跳转至实验的学生列表页面，转第1步； 6a. 系统检测到逻辑分数为负        1. 系统提示逻辑分数必须为正，转第4步； 7a. 系统检测到逻辑分数大于20        1. 系统提示逻辑分数必须小于20，转第4步； 8a. 系统存储实验的逻辑分数时出现系统故障        1. 系统提示评阅逻辑分数失败，转第5步。 | |
| 业务规则 | |
| 1. 教师一次只能评阅一个课程中一个学生的一个实验； 2. 教师填写的逻辑分数必须为正，且小于20； 3. 教师可以重复评阅一个实验的逻辑分数。 | |



# **接口设计**

## **api\_setLogicScore.md（评阅逻辑分数接口）**

* 权限：教师：只能评阅参加自己开设课程的学生的实验逻辑分数，即接口参数test\_id对应的test表中的test.lesson\_id所对应的lesson表中的lesson.teacher\_id必须等于登录教师的teacher\_id，接口参数student\_id必须等于test\_id对应的test表中的test.lesson\_id所对应的choose表中的choose.student\_id或所对应lesson.class\_id必须等于对应student表中的student.class\_id。
* 功能：用于教师评阅逻辑分数
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/test/score/logic/4.set](http://phf449540929.github.io/is_analysis/v1/api/test/score/logic/.set)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"student\_id": "201710414305",

"test\_id": 1,

"logic": "19"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| student\_id | 是 | 学生的学号，对应表student.student\_id的值。 |
| test\_id | 是 | 实验的id，对应表test.test\_id的值。 |
| logic | 是 | 逻辑的分数，对应表score.logic的值。 |

* 返回实例：

{

"information": "评阅逻辑分数成功",

"status": true,

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

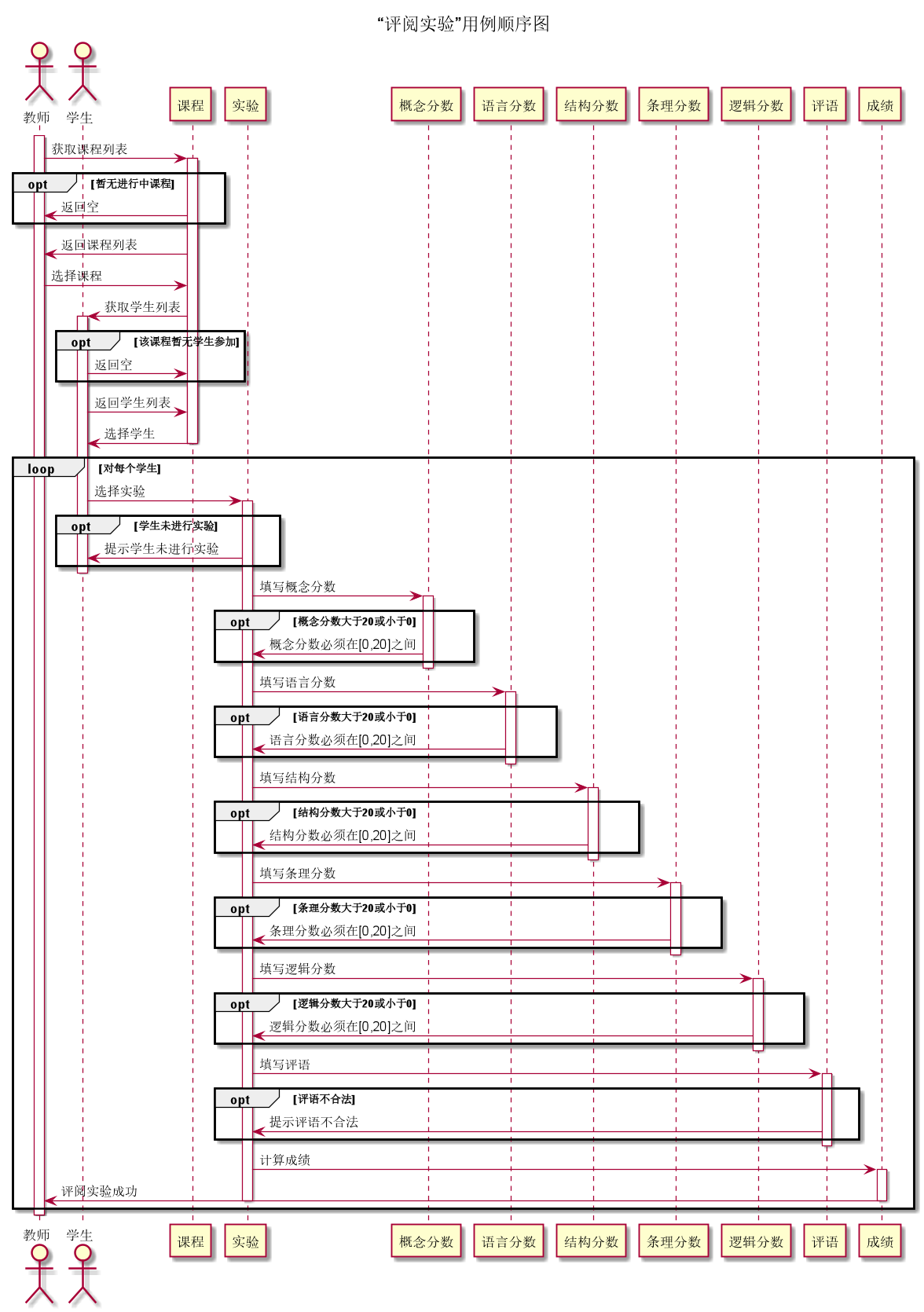


# **用例规约表设计**

## **table\_test\_mark.md（评阅实验用例规约表）**

“评阅实验”用例规约

|  |  |
| --- | --- |
| 用例名称 | 评阅实验 |
| 参与者 | 教师 |
| 前置条件 | 教师已登录，并进入课程实验页面，获取到学生列表 |
| 后置条件 | 系统跳转至实验的学生列表页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 教师选择学生； 教师填写实验的概念分数； 教师填写实验的语言分数； 教师填写实验的结构分数； 教师填写实验的条理分数； 教师填写实验的逻辑分数； .教师填写对该实验的评语； 教师点击提交按钮； | 系统验证教师登录状态是否有效； 系统跳转至实验评分页面； 系统检测评分项分数是否大于0； 系统检测评分项是否小于20； 系统将各评分项成绩相加得实验成绩； 系统存储实验成绩与评语； 系统提示评阅实验成绩成功。 |
| 备选事件流 | |
| 2a. 系统验证教师登录状态已失效        1. 系统提示验证教师登录状态失败，转第1步； 11a.教师选择返回，放弃评阅结构分数        1. 系统跳转至实验的学生列表页面，转第1步； 12a.系统检测到评分项分数为负        1. 系统提示评分项分数必须为正，转第4步； 13a.系统检测到评分项分数大于20        1. 系统提示评分项分数必须小于20，转第4步； 14a.系统存储实验成绩与评语时出现系统故障  1. 系统提示评阅实验成绩失败，转第10步。 | |
| 业务规则 | |
| 1. 教师一次只能评阅一个课程中一个学生的一个实验； 2. 教师填写的评分项分数必须为正，且小于20； 3. 教师可评阅该实验抄袭则实验成绩直接计0； 4. 教师评阅的评分项分数由系统直接计算得实验总成绩； 5. 教师可以重复评阅一个实验的评分项分数。 | |



# **接口设计**

## **api\_setTestScore.md（评阅实验接口）**

* 权限：教师：只能评阅参加自己开设课程的学生的实验成绩，即接口参数test\_id对应的test表中的test.lesson\_id所对应的lesson表中的lesson.teacher\_id必须等于登录教师的teacher\_id，接口参数student\_id必须等于test\_id对应的test表中的test.lesson\_id所对应的choose表中的choose.student\_id或所对应lesson.class\_id必须等于对应student表中的student.class\_id。
* 功能：用于教师评阅实验
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/test/score/.set](http://phf449540929.github.io/is_analysis/v1/api/test/score/.set)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"student\_id": "201710414305",

"test\_id": 1,

"concept": "19",

"language": "19",

"structure": "19",

"method": "19",

"logic": "19",

"plagiarize": false,

"test\_score": "95",

"comment": "做得非常好！"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| student\_id | 是 | 学生的学号，对应表student.student\_id的值。 |
| test\_id | 是 | 实验的id，对应表test.test\_id的值。 |
| concept | 是 | 概念的分数，对应表score.concept的值。 |
| language | 是 | 语言的分数，对应表score.language的值。 |
| structure | 是 | 结构的分数，对应表score.structure的值。 |
| method\_score | 是 | 条理的分数，对应表score.method的值。 |
| logic | 是 | 逻辑的分数，对应表score.logic的值。 |
| plagiarize | 是 | 是否抄袭，true为有抄袭，false为没有抄袭，对应表score.plagiarize。 |
| test\_score | 是 | 实验的总成绩，对应表score.test\_score的值。 |
| comment | 是 | 实验的评语，对应表score.comment的值。 |

* 返回实例：

{

"information": "评阅实验成功",

"status": true,

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

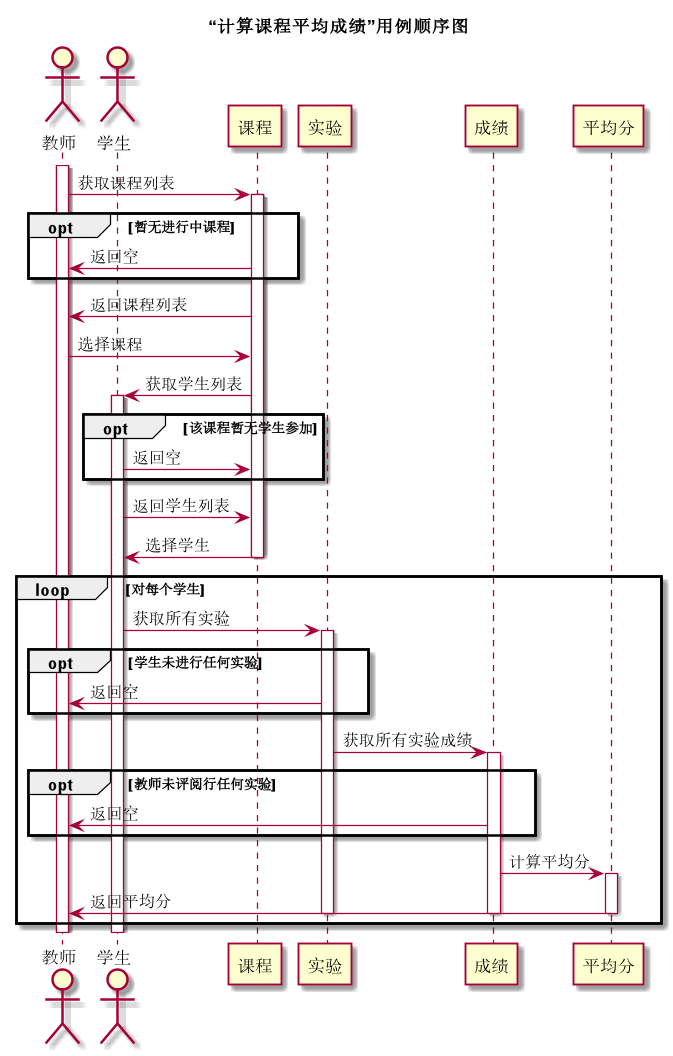


# **用例规约表设计**

## **table\_lesson\_score.md（计算课程平均成绩用例规约表）**

“计算课程平均成绩”用例规约

|  |  |
| --- | --- |
| 用例名称 | 计算课程平均成绩 |
| 参与者 | 教师 |
| 前置条件 | 教师已登录，且已评阅完该课程至少一个实验的成绩 |
| 后置条件 | 系统跳转至实验的学生列表页面 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 教师提交实验的成绩； | 2. 系统验证教师登录状态是否有效； 3. 系统存储实验成绩； 4. 系统检索该学生已被评阅的实验成绩； 5. 系统计算学生已被评阅实验成绩的平均分； 6. 系统存储学生已被评阅实验成绩的平均分； 7. 系统提示计算课程平均成绩成功。 |
| 备选事件流 | |
| 2a. 系统验证教师登录状态已失效        1. 系统提示验证教师登录状态失败，转第1步； 3a. 系统存储实验成绩时出现系统故障        1. 系统提示存储实验成绩失败，转第1步； 4a. 系统检索学生已被评阅的实验成绩时出现系统故障        1. 系统提示检索实验成绩失败，转第1步； 6a. 系统存储课程平均成绩时出现系统故障        1. 系统提示计算课程平均成绩失败，转第1步。 | |
| 业务规则 | |
| 1. 教师评阅各个实验评分项成绩由系统计算平均成绩； 2. 教师可以重复评阅一个实验的评分项分数，系统可以重复计算课程的平均成绩。 | |



# **接口设计**

## **api\_setLessonScore.md（计算课程平均分接口）**

* 权限：教师：只能计算参加自己开设课程的学生的课程分数，即接口参数lesson\_id对应的lesson表中的lesson.teacher\_id必须等于登录教师的teacher\_id，接口参数student\_id必须等于lesson\_id所对应的choose表中的choose.student\_id或所对应lesson.class\_id必须等于对应student表中的student.class\_id。
* 功能：用于教师计算课程平均分
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/lesson/score/.set](http://phf449540929.github.io/is_analysis/v1/api/lesson/score/.set)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"lesson\_id": 11,

"student\_id": "201710414305"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| lesson\_id | 是 | 课程的id，对应表lesson.lesson\_id的值。 |
| student\_id | 是 | 学生的学号，对应表student.student\_id的值。 |

* 返回实例：

{

"information": "维护课程成功",

"status": true,

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

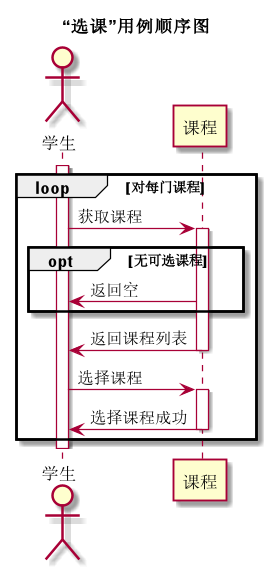


# **用例规约表设计**

## **table\_lesson\_choose.md（选课用例规约表）**

“选课”用例规约

|  |  |
| --- | --- |
| 用例名称 | 选课 |
| 参与者 | 学生 |
| 前置条件 | 学生已登录 |
| 后置条件 | 系统跳转至实验管理平台首页 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 学生点击选课按钮；     6. 学生选择课程； 7. 学生点击提交按钮； | 2. 系统验证学生登录状态是否有效； 3. 系统检索能够被学生选择的课程； 4. 系统返回课程列表； 5. 系统跳转至选课页面；   8. 系统存储学生选择的课程； 9. 系统提示选课成功。 |
| 备选事件流 | |
| 2a. 系统验证学生登录状态已失效        1. 系统提示验证学生登录状态失败，转第1步； 4a. 系统未能检索到能够被学生选择的课程        1. 系统提示无可选课程，转第1步； 4b. 系统在检索课程时出现系统故障        1. 系统提示检索课程失败，转第1步； 7a. 学生选择返回，放弃选课        1. 系统跳转至实验管理平台首页，转第1步； 8a. 系统存储选课结果时出现系统故障        1. 系统提示提交选课失败，转第7步。 | |
| 业务规则 | |
| 1. 学生一次只能选择一个课程； 2. 学生只能选择其能够选择的课程。 | |



# **接口设计**

## **api\_chooseLesson.md（选课级接口）**

* 权限：学生：只能给自己选课，即接口参数student\_id必须等于登录学生的student\_id。
* 功能：用于学生选课
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/lesson/.choose](http://phf449540929.github.io/is_analysis/v1/api/lesson/.choose)
* 请求方法：POST
* 请求实例：

{

"access\_token": $access\_token,

"method": "POST",

"student\_id": "201710414305",

"lesson\_id": 11

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“POST”。 |
| student\_id | 是 | 学生的学号，对应表student.student\_id的值。 |
| lesson\_id | 是 | 课程的id，对应表lesson.lesson\_id的值。 |

* 返回实例：

{

"information": "选课成功",

"status": true,

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| code | 是 | 返回码。 |

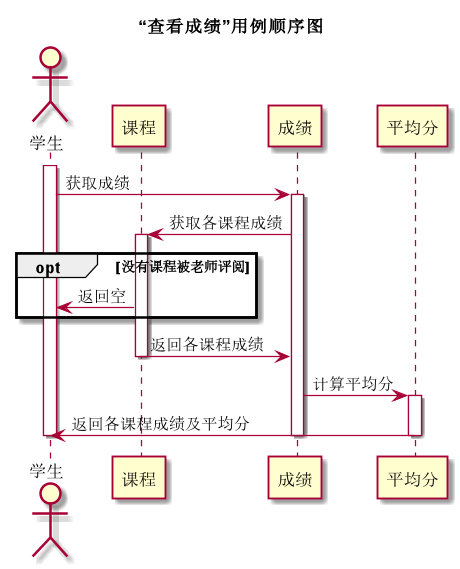


# **用例规约表设计**

## **table\_student\_score.md（查看成绩用例规约表）**

“查看成绩”用例规约

|  |  |
| --- | --- |
| 用例名称 | 查看成绩 |
| 参与者 | 学生 |
| 前置条件 | 学生已登录 |
| 后置条件 | 系统跳转至实验管理平台首页 |
| 主事件流 | |
| 参与者动作 | 系统行为 |
| 1. 学生点击查看成绩按钮； | 2. 系统验证学生登录状态是否有效； 3. 系统检索学生的各课程成绩； 4. 系统计算学生的各课程平均成绩； 5. 系统返回该学生的各课程成绩及平均成绩。 |
| 备选事件流 | |
| 2a. 系统验证学生登录状态已失效        1. 系统提示验证学生登录状态失败，转第1步； 3a. 系统未检索到学生各课程成绩        1. 系统提示暂未进行实验或教师暂未评阅，转第1步； 3b. 系统检索学生各课程成绩时出现系统故障        1. 系统提示检索学生成绩失败，转第1步。 | |
| 业务规则 | |
| 1. 学生可以查看其它学生的成绩， 2. 学生的各课程平均成绩由系统自动计算。 | |



# **接口设计**

## **api\_setStudentScore.md（查看成绩接口）**

* 权限：学生：只能查看自己的成绩，即接口参数student\_id必须等于登录学生的student\_id。
* 功能：用于学生查看成绩
* 请求地址：[http://NamelessSoldier.github.io/is\_analysis/v1/api/student/score/.get](http://phf449540929.github.io/is_analysis/v1/api/student/score/.get)
* 请求方法：GET
* 请求实例：

{

"access\_token": $access\_token,

"method": "GET",

"student\_id": "201710414305"

}

* 请求参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| access\_token | 是 | 用于验证请求合法性的认证信息。 |
| method | 是 | 固定为“GET”。 |
| student\_id | 是 | 学生的学号，对应表student.student\_id的值。 |

* 返回实例：

{

"information": "查看成绩成功",

"status": true,

"average": 75,

"test\_score\_list": [  
 [  
 {"name": "概念分数", "test\_score":"99"},  
 {"name": "结构分数", "test\_score":"99"},  
 {"name": "逻辑分数", "test\_score":"99"},  
 {"name": "语言分数", "test\_score":"99"},  
 {"name": "条理分数", "test\_score":"99"}  
 ]  
 [  
 {"name": "xxx", "test\_score":"99"},  
 {"name": "xxx", "test\_score":"99"},  
 {"name": "xxx", "test\_score":"99"},  
 {"name": "xxx", "test\_score":"99"},  
 {"name": "xxx", "test\_score":"99"}  
 ]

],

"code": 200

}

* 返回参数说明：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 必填 | 说明 |
| information | 是 | 返回的说明信息。 |
| status | 是 | 返回的结果，boolean类型，true表示正确的返回，false表示有错误。 |
| average | 否 | 学生所有课程的平均成绩，对应表student.student\_average的值。 |
| lesson\_score\_list | 否 | 学生所有课程的平均成绩成绩列表。 |
| test\_score\_list | 否 | 学生所有课程的所有实验的成绩列表。 |
| code | 是 | 返回码。 |

