

**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 | 陈佳豪 | 学号 | 201710414201 |
| 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

目录

[1.概述 3](#_Toc7971475)

[2.总体结构 3](#_Toc7971476)

[3.用例图设计 4](#_Toc7971477)

[4.类图设计 5](#_Toc7971479)

[5.数据库设计 7](#_Toc7971480)

[ **users表（用户表）** 7](#_Toc7971481)

[ **teacher表（老师表）** 7](#_Toc7971482)

[ **student表（学生表）** 7](#_Toc7971483)

[ **courses表（课程汇总表）** 8](#_Toc7971484)

[ **course表（课程表）** 8](#_Toc7971485)

[ **test表（实验表）** 8](#_Toc7971486)

[ **grade表（成绩表）** 9](#_Toc7971487)

[6.用例及界面详细设计 9](#_Toc7971488)

[“学生列表” 9](#_Toc7971489)

[“评定成绩”用例 14](#_Toc7971494)

[“登录”用例 18](#_Toc7971501)

[“登出”用例 22](#_Toc7971506)

[“选择学期”用例 24](#_Toc7971511)

[“选择课程”用例 27](#_Toc7971516)

[“修改密码”用例 30](#_Toc7971521)

[“修改用户信息”用例 33](#_Toc7971526)

[“查看用户信息”用例 37](#_Toc7971531)

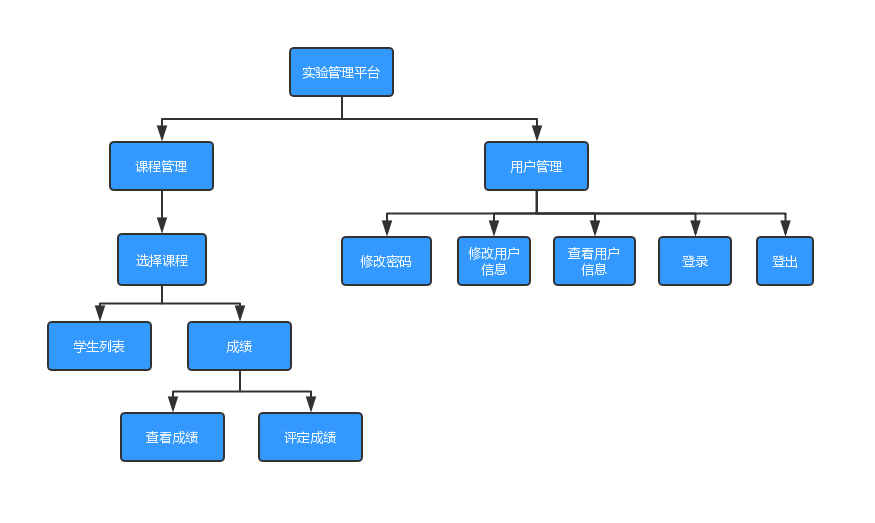
[“查看成绩”用例 39](#_Toc7971536)

**实验六：基于GitHub的实验管理平台的分析与设计**

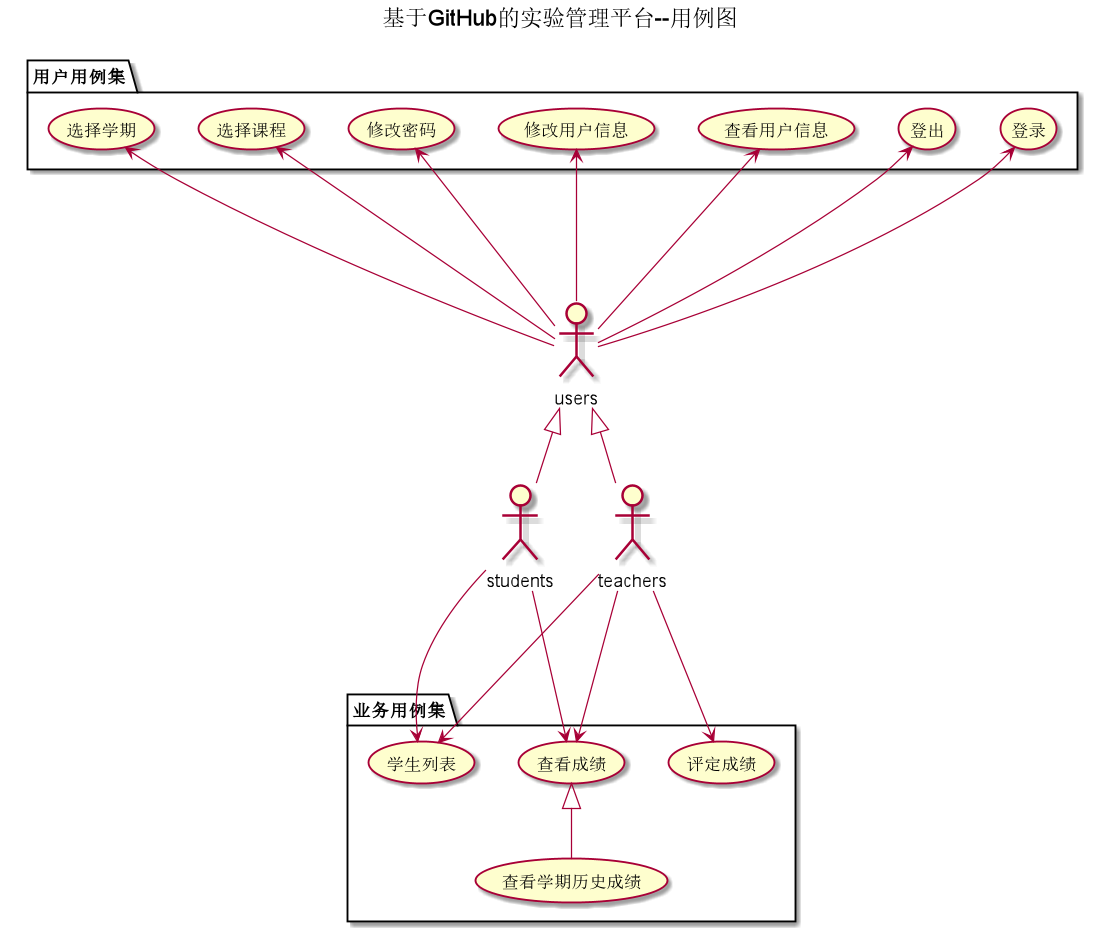
### 1.概述

* 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。学生和老师的实验内容均存放在GitHUB 页面上。
* 学生的功能主要有：一是设置自己的GitHub用户名，二是选择课程，三是查询自己的每一门课程的实验成绩。学生的GitHub用户名是公开的，但成绩不公开。
* 老师的功能主要有：一是选择课程，二是批改每个学生的成绩，三是查看每个学生每个课程的成绩。
* 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.总体结构



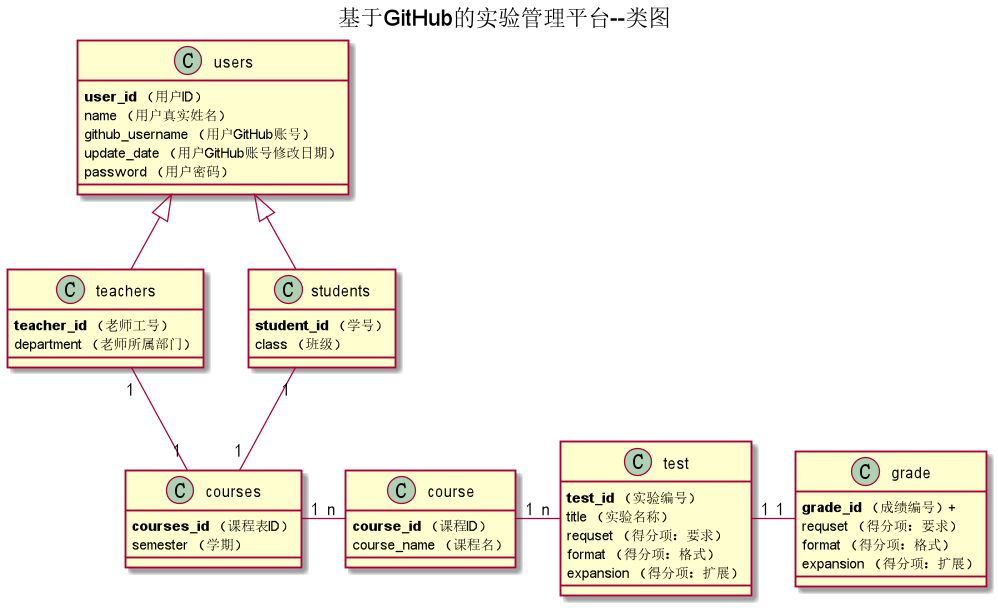
### 3.用例图设计



### 源码：

|  |
| --- |
| @startuml |
|  | title 基于GitHub的实验管理平台--用例图 |
|  | actor teachers |
|  | actor students |
|  | actor users |
|  | users <|-- teachers |
|  | users <|-- students |
|  | package 用户用例集 { |
|  | users --up-> (登录) |
|  | users --up-> (登出) |
|  | users --up-> (查看用户信息) |
|  | users --up-> (修改用户信息) |
|  | users --up-> (修改密码) |
|  | users --up-> (选择课程) |
|  | users --up-> (选择学期) |
|  | } |
|  |  |
|  | package 业务用例集 { |
|  | (查看学期历史成绩)-up-|>(查看成绩) |
|  | teachers ---> (评定成绩) |
|  | teachers ---> (学生列表) |
|  | teachers-->(查看成绩) |
|  | students ---> (学生列表) |
|  | students ---> (查看成绩) |
|  | } |
|  | @enduml |

### 4.类图设计



源码：

|  |
| --- |
| @startuml |
|  | title 基于GitHub的实验管理平台--类图 |
|  | class users { |
|  | <b>user\_id</b> （用户ID） |
|  | name （用户真实姓名） |
|  | github\_username （用户GitHub账号） |
|  | update\_date （用户GitHub账号修改日期） |
|  | password （用户密码） |
|  | } |
|  | class teachers{ |
|  | <b>teacher\_id</b> （老师工号） |
|  | department （老师所属部门） |
|  | } |
|  | class students{ |
|  | <b>student\_id</b> （学号） |
|  | class （班级） |
|  | } |
|  | users <|-- students |
|  | users <|- teachers |
|  | class courses{ |
|  | <b>courses\_id</b> （课程表ID） |
|  | semester （学期） |
|  | } |
|  |  |
|  | class course{ |
|  | <b>course\_id</b> （课程ID） |
|  | course\_name （课程名） |
|  | } |
|  | class test { |
|  | <b>test\_id</b> （实验编号） |
|  | title （实验名称） |
|  | requset （得分项：要求） |
|  | format （得分项：格式） |
|  | expansion （得分项：扩展） |
|  | } |
|  | class grade{ |
|  | <b>grade\_id</b> （成绩编号）+ |
|  | requset （得分项：要求） |
|  | format （得分项：格式） |
|  | expansion （得分项：扩展） |
|  | } |
|  |  |
|  | students "1" -- "1" courses |
|  | courses "1" - "n" course |
|  | course "1" - "n" test |
|  | teachers "1" - "1" courses |
|  | test "1" - "1" grade |
|  |  |
|  | @enduml |

### 5.数据库设计

* **users表（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(10) | 主键 | 否 | 数据库自增 |  | 用户 |
| Name | varchar（50） |  | 否 |  |  | 真实姓名 |
| github\_username | varchar（50） |  | 是 | 空 |  | github用户名 |
| update\_date | date |  | 是 | 空 |  | 修改日期 |
| Password | varchar（50） |  | 否 | 123456 |  | 登陆密码 |
| Disable | varchar（20） |  | 否 |  |  | 是否禁用 |

* **teacher表（老师表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| teccher\_id | varchar（50） | 主键 | 否 |  |  | 老师编号 |
| user\_id | varchar(50) | 外键 | 否 |  | 参见users表主键 | 用户ID |
| departemt | varchar（50） |  | 否 |  |  | 老师部门 |

* **student表（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 |  |  | 学生学号 |
| user\_id | varchar(50) | 外键 | 否 |  | 参见users表主键 | 学生用户ID |
| major | varchar（20） |  | 否 |  |  | 学生专业 |
| classno | varchar(20) |  | 否 |  |  | 学生班级 |

* **courses表（课程汇总表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| courses\_id | varchar(50) | 主键 | 否 |  |  | 学生个人课程表ID |
| semester | varchar（20） |  | 否 |  |  | 课程所属学期 |
| student\_id | varchar(50) | 外键 | 否 |  | 参见student表主键 | 学生ID |
| teacher\_id | varchar(50) | 外键 | 否 |  | 参见teacher表主键 | 老师ID |

* **course表（课程表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| course\_id | varchar(50) | 主键 | 否 |  |  | 唯一标识一门课程 |
| course\_name | varchar（20） |  | 否 |  |  | 课程名 |
| courses\_id | varchar(50) | 外键 | 否 |  | 参见ourses表主键 | 标识所属课程表 |

* **test表（实验表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| test\_id | varchar(50) | 主键 | 否 |  |  | 唯一标识一次实验 |
| Title | varchar(20) |  | 否 |  |  | 实验名 |
| request | varchar(255) |  | 否 |  |  | 得分项：要求 |
| format | varchar(255) |  | 否 |  |  | 得分项：格式 |
| expansion | varchar(255) |  | 否 |  |  | 得分项：扩展 |
| course\_id | varchar(50) | 外键 | 否 |  | 参见course表主键 | 标识所属课程 |

* **grade表（成绩表）**

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| grade\_id | varchar(50) | 主键 | 否 |  |  | 唯一标识一个test成绩 |
| Title | varchar(20) |  | 否 |  |  | 实验名 |
| request | int(10) |  | 否 |  |  | 得分项：要求得分 |
| format | int(10) |  | 否 |  |  | 得分项：格式得分 |
| expansion | int(10) |  | 否 |  |  | 得分项：扩展得分 |
| comment | varchar(255) |  | 是 |  |  | 评语 |
| test\_id | varchar(50) | 外键 | 否 |  | 参见test表主键 | 标识所属test |

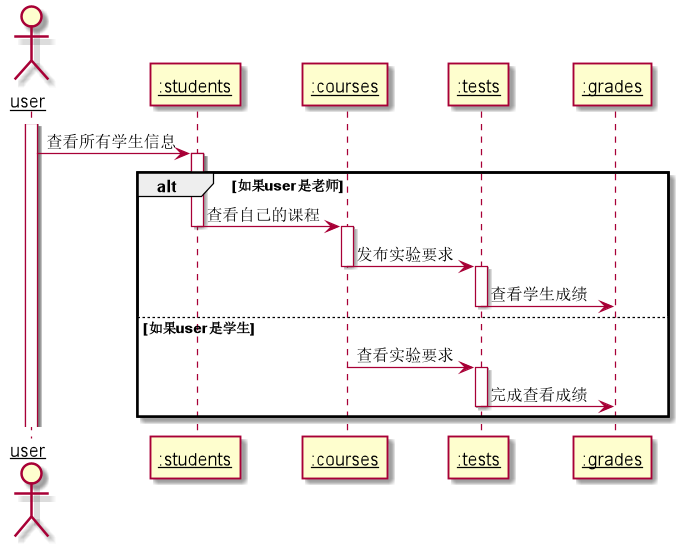
### 6.用例及界面详细设计

# “学生列表”

## 1. 用例规约

| **用例名称** | **学生列表** |
| --- | --- |
| 功能 | 显示所有学生信息 |
| 参与者 | 学生，老师,未注册用户 |
| 前置条件 | 学生老师均需要先登录系统 |
| 后置条件 | 学生和老师可以开始使用实验系统完成教学任务 |
| 主事件流 | 1.学生登录  2.老师登录  3.游客路过 |
| 备选事件流 | 1.登录失败 2.网络错误 |

## 2. 业务流程（顺序图）



源码：

|  |
| --- |
| @startuml |
|  | skinparam sequenceParticipant underline |
|  | actor user as user |
|  | participant ":students" as students |
|  | participant ":courses" as courses |
|  | participant ":tests" as tests |
|  | participant ":grades" as grades |
|  | activate user |
|  | user->students:查看所有学生信息 |
|  | alt 如果user是老师 |
|  | activate students |
|  | students->courses:查看自己的课程 |
|  | deactivate students |
|  | activate courses |
|  | courses->tests:发布实验要求 |
|  | deactivate courses |
|  | activate tests |
|  | tests->grades:查看学生成绩 |
|  | deactivate tests |
|  | else 如果user是学生 |
|  | courses->tests:查看实验要求 |
|  | activate tests |
|  | tests->grades:完成查看成绩 |
|  | deactivate tests |
|  | end |
|  | @enduml |

## 3. 界面设计

**API接口调用**

* 接口：getStudents
* 权限： 游客只能看到学生列表信息  
  学生不能看到老师发布实验要求的页面
* 功能： 返回所有学生的列表。
* API请求地址： 接口基本地址/v1/api/getStudents
* 请求方式 ： GET
* 请求参数说明: 无
* 返回实例：

{

"status": true,

"info": null,

"total": 121,

"data": [

{

"student\_id":"201710414201",

"github\_username":"SakaiYoui",

"class":"软件（本）2017-2",

"name":"陈佳豪",

"update\_date":"2019-04-25"

},

{

...其他学生

}

]

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| status | true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |
| total | 返回学生人数 |
| data | 所有学生的数组 |
| github\_username | github 用户名 |
| student\_id | 学号 |
| class | 班级 |
| NAME | 真实姓名 |
| update\_date | GitHUB用户名修改日期 |

## 4. 参照表

* [student](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#student)
* [courses](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#courses)
* [test](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#test)
* [grade](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#grade)

# “评定成绩”用例

## 1. 用例规约

| **用例名称** | **评定成绩** |
| --- | --- |
| 功能 | 老师评定一个学生的实验成绩 |
| 参与者 | 系统中已注册的并且有选课的老师 |
| 前置条件 | 1.老师先登录，并且实验要求已经发布，学生已经提交 2.显示实验评分项，及评语 |
| 后置条件 | 1.成绩评定完成后保存到课程成绩中， 2.系统通过每一次实验的成绩统计出平均成绩显示出来 |
| 主事件流 | 1. 查看已有的成绩  2. 根据得分项给出得分和评语  3. 提交  4. 系统存储实验成绩和评语 5. 系统自动计算平均成绩 |
| 备选事件流 | 1.每一个得分项的成绩必须合法，可以为空，但不能超界     1.提示用户重新输入成绩分数 |

## 评定成绩.png2. 业务流程（顺序图）

## 

## 源码：

|  |
| --- |
| @startuml |
|  | skinparam sequenceParticipant underline |
|  | actor teacher |
|  | participant ":course" as course |
|  | participant ":students" as students |
|  | participant ":test" as test |
|  | participant ":grade" as grade |
|  | activate teacher |
|  | loop |
|  | teacher ->course:选择课程 |
|  | loop |
|  | course->students:选择批改学生 |
|  | students ->test:查阅学生实验 |
|  | test->grade:给出成绩 |
|  | end |
|  | end |
|  | @enduml |

## 3. 界面设计

**API接口调用**

接口：getStudents [返回](https://github.com/zemaochen/is_analysis/blob/master/test6/README.md)

* 权限： 游客只能看到学生列表信息  
  学生不能看到老师发布实验要求的页面
* 功能： 返回所有学生的列表。
* API请求地址： 接口基本地址/v1/api/getStudents
* 请求方式 ： GET
* 请求参数说明: 无
* 返回实例：

{

"status": true,

"info": null,

"total": 121,

"data": [

{

"student\_id":"201710414201",

"github\_username":"SakaiYouji",

"class":"软件（本）2016-1",

"name":"陈佳豪",

"update\_date":"2019-04-25"

},

{

...其他学生

}

]

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| status | true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |
| total | 返回学生人数 |
| data | 所有学生的数组 |
| github\_username | github 用户名 |
| student\_id | 学号 |
| class | 班级 |
| NAME | 真实姓名 |
| update\_date | GitHUB用户名修改日期 |

接口：setOneStudentResults

* 功能： 老师根据得分项评分
* 权限： 老师：可以查看所有学生的实验。
* API请求地址： 接口基本地址/v1/api/etOneStudentResults
* 请求方式 ： POST
* 请求实例：

{

"student\_id": "201710414201",

"status": true,

"info": null,

"data":

{

"grade\_id":"01",

"title":"实验一",

"request":48,

"format":18,

"expansion":15,

"comment":"本次实验做的不错"

}

}

* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| student\_id | 学号 |
| data | 实验的各得分项得分 |
| grade\_id | 成绩单编号 |
| comment | 本实验老师的评价，可以为空 |

* 返回实例：

{

"status": true,

"info": null

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| Status | bool类型，true表示正确的返回，false表示有错误 |
| Info | 返回结果说明信息 |

## 4. 参照表

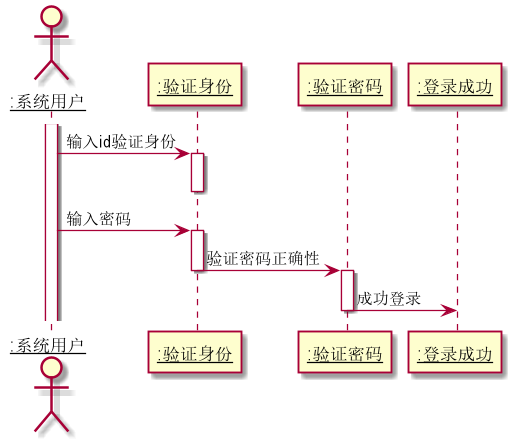
* [student](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#student)
* [grade](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#grade)
* [test](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#test)
* [course](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#course)

# “登录”用例

## 1. 用例规约

| **用例名称** | **用户登录** |
| --- | --- |
| 参与者 | 老师或学生 |
| 前置条件 | 用户已在在系统中注册 |
| 后置条件 | 登录成功，记录登录信息（日志） |
| 主事件流 | 1.用户输入用户名和密码 2.系统验证用户身份 3.系统验证用户密码 4.认证通过 |
| 备注 | 用户登录输入用户名系统查询是否有该用户存在，验证密码后登录成功 |
| 备选事件流 | a.用户不存在：提示id输入错误 b.密码错误：1.提示密码错误重新输入 |

## 2. 业务流程（顺序图）



**源码：**

|  |
| --- |
| @startuml |
|  | skinparam sequenceParticipant underline |
|  | actor ":系统用户" as A |
|  | participant ":验证身份" as B |
|  | participant ":验证密码" as C |
|  | participant ":登录成功" as D |
|  | activate A |
|  | A->B:输入id验证身份 |
|  | activate B |
|  | deactivate |
|  | A->B:输入密码 |
|  | activate B |
|  | B->C:验证密码正确性 |
|  | deactivate |
|  | activate C |
|  | C->D:成功登录 |
|  | deactivate |
|  | @enduml |

## 3. 界面设计

**API接口调用**

* 登录
* 功能：用于用户的登录
* 请求地址：http://[yourname]/api/login
* 请求方法：POST
* 请求参数：

| **参数名称** | **必填** | **说明** |
| --- | --- | --- |
| user\_id | 是 | 用户ID |
| Password | 是 | 密码 |
| Method | 是 | 固定为"POST" |

* 返回示例：

{

"data":

{

"info":"返回信息",

"user":

{

"user\_id":"1",

"name":"小李",

"last\_login\_time":"1438329358"

}

},

"code":200

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| Data | 返回参数数据主体 |
| Code | 返回码 |

## 4. 参照表

* [user](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#user)

# “登出”用例

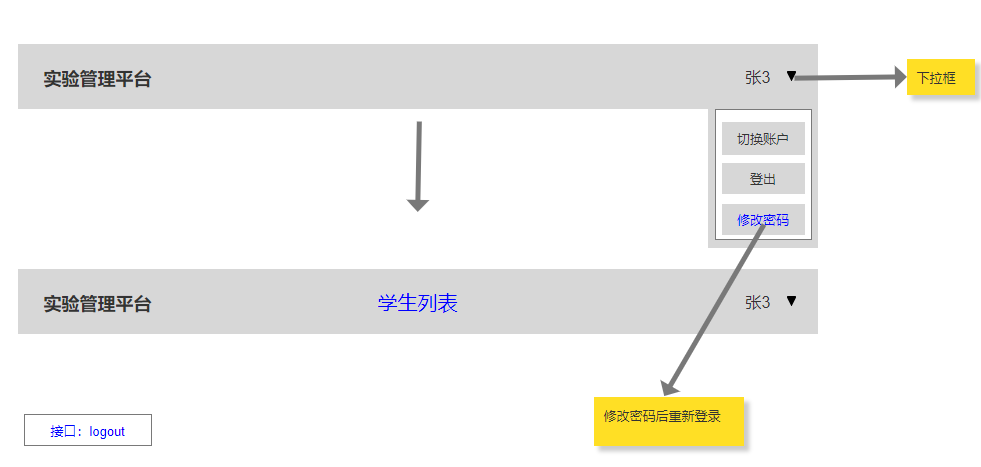
## 1. 用例规约

| **用例名称** | **登出** |
| --- | --- |
| 功能 | 用户登出平台 |
| 参与者 | 学生/老师 |
| 前置条件 | 用户已登录 |
| 后置条件 | 登出后，跳转到登录页面 |
| 主事件流 | 1. 系统清除客户端登录信息，退出系统 |
| 备选事件流 | 1. 如果用户登录之后，长时间不超作界面，导致Cookie失效     1.系统清除客户端登录信息（Cookie |

## 2. 业务流程

无

## 3. 界面设计

* 界面参照: [https://SakaiYouji.github.io/is\_analysis\_pages/test6/bar.html](https://zemaochen.github.io/is_analysis_pages/test6/bar.html)

**API接口调用**

* + 接口：[logout](https://github.com/zemaochen/is_analysis/blob/master/test6/%E6%8E%A5%E5%8F%A3/logout.md)

接口：logout

用例： [登出](https://github.com/zemaochen/is_analysis/blob/master/test6/%E7%94%A8%E4%BE%8B/%E7%99%BB%E5%87%BA.md)

* 功能： 已经登录的用户登出平台。
* 权限： 学生/老师已经登录。
* API请求地址： 接口基本地址/v1/api/logout
* 请求方式 ： POST
* 请求实例：

{

"user\_id":123456

}

* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| user\_id | 已经登录用户的user\_id值，对应表user的user\_id |

* 返回实例：

{

"status": true,

"info": null

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| Status | bool类型，true表示正确的返回，false表示有错误 |
| Info | 返回结果说明信息 |

## 4. 参照表

* [user](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#user)

# “选择学期”用例

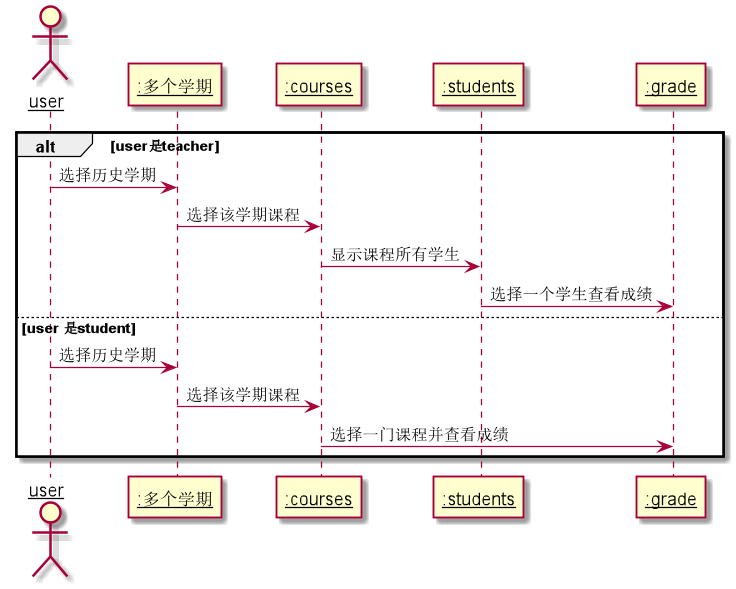
## 1. 用例规约

| **用例名称** | **选择学期** |
| --- | --- |
| 功能 | 选择学期，查询历史成绩 |
| 参与者 | 学生，老师 |
| 前置条件 | 1.老师学生先登录 2.不止一个学期 |
| 后置条件 | 显示多个学期供用户选择 |
| 主事件流 | 1.选择一个学期 |
| 备选事件流 | 只有一个学期，没有选择 |

## 2. 业务流程（顺序图）

源码：

|  |
| --- |
| @startuml |
|  | skinparam sequenceParticipant underline |
|  | actor user |
|  | participant ":多个学期" as semesters |
|  | participant ":courses" as courses |
|  | participant ":students" as students |
|  | participant ":grade" as grade |
|  | alt user是teacher |
|  | user->semesters:选择历史学期 |
|  | semesters->courses:选择该学期课程 |
|  | courses->students:显示课程所有学生 |
|  | students->grade:选择一个学生查看成绩 |
|  | else user 是student |
|  | user->semesters:选择历史学期 |
|  | semesters->courses:选择该学期课程 |
|  | courses->grade:选择一门课程并查看成绩 |
|  | end |
|  | @enduml |



## 3. 界面设计

**API接口调用**

* 接口：getSemester
* 功能： 取得历史学期
* 权限： 学生/老师已经登录。
* API请求地址： 接口基本地址/v1/api/getSemester
* 请求方式 ： POST
* 请求实例：

{

"user\_id": "123456"

}

* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| user\_id | 已经登录用户的user\_id值，对应表user的user\_id |

* 返回实例：

{

"status": true,

"info": null,

"data":[

{

"semester": "xx学年第一学期"

},

{

...其他学期

}

]

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| Status | bool类型，true表示正确的返回，false表示有错误 |
| Info | 返回结果说明信息 |
| Data | 返回学期数组信息 |

## 4. 参照表

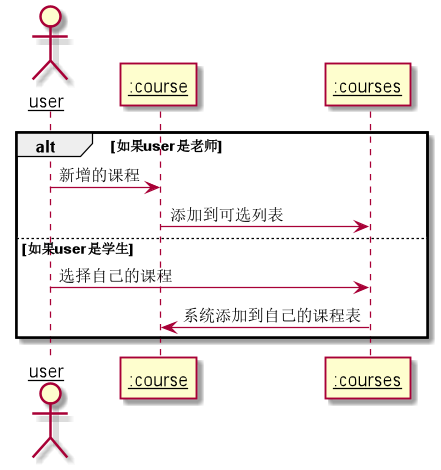
* [student](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#student)
* [teacher](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#teacher)
* [courses](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#courses)

# “选择课程”用例

## 1. 用例规约

| **用例名称** | **选择课程** |
| --- | --- |
| 功能 | 选课 |
| 参与者 | 学生，老师 |
| 前置条件 | 1.老师学生先登录 |
| 后置条件 | 选择课程后学生拥有该课程，老师管理该课程 |
| 主事件流 | 老师先选择课程，学生后选择课程 |
| 备选事件流 |  |

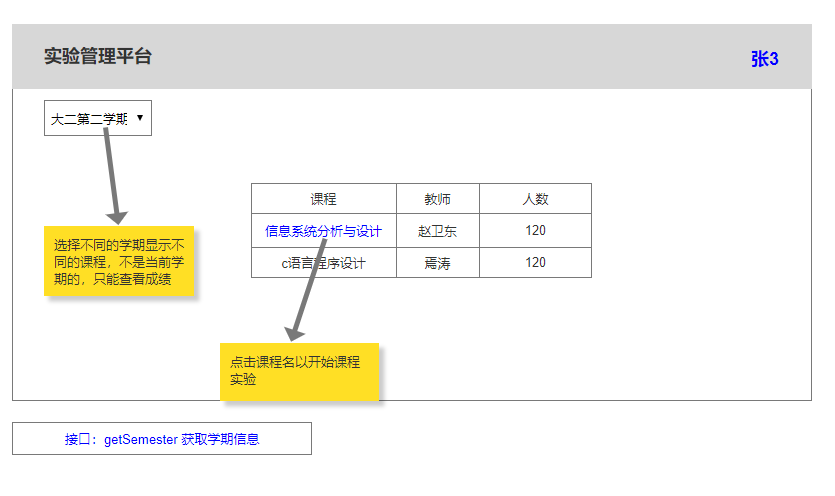
## 2. 业务流程（顺序图）



源码：

|  |
| --- |
| @startuml |
|  | skinparam sequenceParticipant underline |
|  | actor user |
|  | participant ":course" as course |
|  | participant ":courses" as courses |
|  | alt 如果user是老师 |
|  | user->course:新增的课程 |
|  | course->courses:添加到可选列表 |
|  | else 如果user是学生 |
|  | user->courses:选择自己的课程 |
|  | courses->course:系统添加到自己的课程表 |
|  | end |
|  | @enduml |

## 3. 界面设计



**API接口调用：**

* 接口：getSemester
* 功能： 取得历史学期
* 权限： 学生/老师已经登录。
* API请求地址： 接口基本地址/v1/api/getSemester
* 请求方式 ： POST
* 请求实例：

{

"user\_id": "123456"

}

* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| user\_id | 已经登录用户的user\_id值，对应表user的user\_id |

* 返回实例：

{

"status": true,

"info": null,

"data":[

{

"semester": "xx学年第一学期"

},

{

...其他学期

}

]

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |
| data | 返回学期数组信息 |

## 4. 参照表

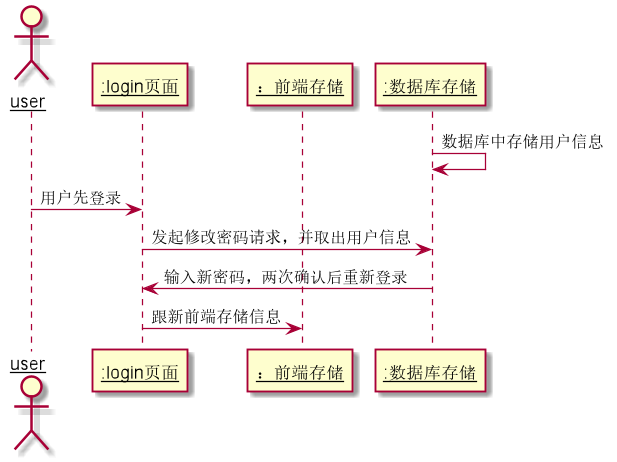
* [user](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#user)
* [student](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#student)
* [teacher](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#teacher)
* [courses](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#courses)

# “修改密码”用例

## 1. 用例规约

| **用例名称** | **修改密码** |
| --- | --- |
| 功能 | 修改用户的密码 |
| 参与者 | 学生，老师 |
| 前置条件 | 用户必须先登录，并提供原始密码 |
| 后置条件 | 修改密码以后必须强制登出，再跳转到登录页面 |
| 主事件流 | 1.用户填写新密码（两次输入）  2.用户提交修改信息  3.系统存储修改后密码 |
| 备选事件流 | 1a. 用户两次输入的密码不同     1.系统提示两次输入的密码不相同     2. 用户重新填写并提交。 |

## 2. 业务流程



源码：

|  |
| --- |
| @startuml |
|  | skinparam sequenceParticipant underline |
|  | actor user |
|  | participant ":login页面" as login页面 |
|  | participant "：前端存储" as 前端 |
|  | participant ":数据库存储" as 数据库 |
|  | 数据库->数据库:数据库中存储用户信息 |
|  | user->login页面:用户先登录 |
|  | login页面->数据库:发起修改密码请求，并取出用户信息 |
|  | 数据库->login页面:输入新密码，两次确认后重新登录 |
|  | login页面->前端:跟新前端存储信息 |
|  |  |
|  | @enduml |

## 3. 界面设计



**API接口调用：**

* 接口：setPassword
* 功能： 修改（设置）用户的密码。
* 权限： 学生/老师：修改自己的密码，必须先登录。
* API请求地址： 接口基本地址/v1/api/setPassword
* 请求方式 ： POST
* 请求实例：

{

"user\_id":1234,

"password":"ADDFHLIDCNPQWECE"

}

* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| user\_id | 用户id |
| password | 用户的密码。加密传输 |

* 返回实例：

{

"status": true,

"info": null,

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |

## 4. 参照表

users

* [user](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#user)

# “修改用户信息”用例

## 1. 用例规约

| **用例名称** | **修改用户信息** |
| --- | --- |
| 功能 | 修改用户的GitHub用户名称 |
| 参与者 | 学生，老师 |
| 前置条件 | 登录并查看用户现在的github信息 |
| 后置条件 | 系统发送修改请求，数据信息跟新 |
| 主事件流 | 1.用户填写GitHub用户名称  2.用户提交修改信息  3.系统存储修改后的GitHub用户名称 |
| 备选事件流 | 1a. 如果用户输入的GitHub用户名称为空     1.系统清空用户的GitHub用户名称 |

## 2. 业务流程

无

## 3. 界面设计



* API接口调用
* 接口：getUserInfo
* 功能： 查看用户的所有信息。
* 权限： 学生/老师：查看自己的信息，必须先登录，不能查看其他用户的信息。
* API请求地址： 接口基本地址/v1/api/getUserInfo/<user\_id>
* 请求方式 ： GET
* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| user\_id | 用户的ID号。对应表USERS.USER\_ID的值 |

* 返回实例：

{

"status": true,

"info": null,

"data": {

"user\_id":"2101201281",

"name":"张三",

"github\_username":"ABCD"

}

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |
| data | 返回数据实体 |

* 接口：setUserInfo
* 功能： 修改用户的GitHub用户名。
* 权限： 学生/老师：修改自己的密码，必须先登录。
* API请求地址： 接口基本地址/v1/api/setUserInfo
* 请求方式 ： POST
* 请求实例：

{

"user\_id":"21048329823",

"github\_username":"ABCDE"

}

* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| user\_id | 用户id |
| github\_username | 要修改的github名 |

* 返回实例：

{

"status": true,

"info": null

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |

## 4. 参照表

* [user](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#user)

# “查看用户信息”用例

## 1. 用例规约

| **用例名称** | **查看用户信息** |
| --- | --- |
| 功能 | 查看用户的所有信息 |
| 参与者 | 学生，老师 |
| 前置条件 | 必须先登录 |
| 后置条件 |  |
| 主事件流 |  |
| 备选事件流 |  |

## 2. 业务流程

无

## 3. 界面设计

**API接口调用：**

* 接口：getUserInfo
* 功能： 查看用户的所有信息。
* 权限： 学生/老师：查看自己的信息，必须先登录，不能查看其他用户的信息。
* API请求地址： 接口基本地址/v1/api/getUserInfo/<user\_id>
* 请求方式 ： GET
* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| user\_id | 用户的ID号。对应表USERS.USER\_ID的值 |

* 返回实例：

{

"status": true,

"info": null,

"data": {

"user\_id":"2101201281",

"name":"张三",

"github\_username":"ABCD"

}

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |
| data | 返回数据实体 |

## 4. 参照表

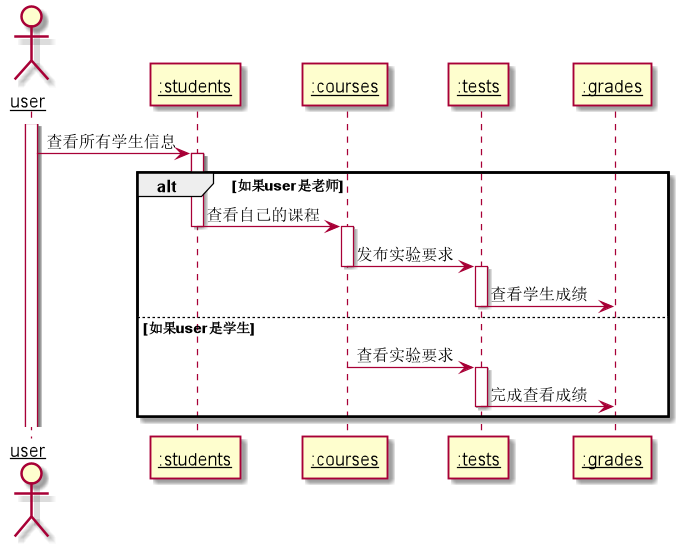
* [student](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#student)
* [teacher](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#teacher)
* [user](https://github.com/zemaochen/is_analysis/blob/master/test6/md/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#user)

# “查看成绩”用例

## 1. 用例规约

| **用例名称** | **查看成绩** |
| --- | --- |
| 功能 | 学生查看自己的每个实验的实验成绩及实验评价 |
| 参与者 | 学生 |
| 前置条件 | 学生需要先登录 |
| 后置条件 |  |
| 主事件流 |  |
| 备选事件流 |  |

## 2. 业务流程（顺序图）



源码：

|  |
| --- |
| @startuml |
|  | skinparam sequenceParticipant underline |
|  | actor user as user |
|  | participant ":students" as students |
|  | participant ":courses" as courses |
|  | participant ":tests" as tests |
|  | participant ":grades" as grades |
|  | activate user |
|  | user->students:查看所有学生信息 |
|  | alt 如果user是老师 |
|  | activate students |
|  | students->courses:查看自己的课程 |
|  | deactivate students |
|  | activate courses |
|  | courses->tests:发布实验要求 |
|  | deactivate courses |
|  | activate tests |
|  | tests->grades:查看学生成绩 |
|  | deactivate tests |
|  | else 如果user是学生 |
|  | courses->tests:查看实验要求 |
|  | activate tests |
|  | tests->grades:完成查看成绩 |
|  | deactivate tests |
|  | end |
|  | @enduml |

## 3. 界面设计API接口调用：

* 接口：getOneStudentResults
* 功能： 返回一个学生的所有实验成绩和实验评价。
* 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 老师：可以查看所有学生的成绩。
* API请求地址： 接口基本地址/v1/api/getOneStudentResults/<student\_id>
* 请求方式 ： GET
* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| student\_id | 学生的学号 |

* 返回实例：

{

"student\_id": "201710414201",

"status": true,

"info": null,

"data":

{

"grade\_id":"01",

"title":"实验一",

"request":48,

"format":18,

"expansion":15,

"comment":"本次实验做的不错"

}

}

* 返回参数说明：  
  |参数名称|说明| |:---------:|:--------------------------------------------------------|  
  |student\_id|学号| |data|实验的各得分项得分| |grade\_id|成绩单编号| |comment|本实验老师的评价，可以为空|
* 返回实例：

{

"status": true,

"info": null

}

* 返回参数说明

| **参数名称** | **说明** |
| --- | --- |
| student\_id | 学号 |
| data | 实验的各得分项得分 |
| grade\_id | 成绩单编号 |
| comment | 本实验老师的评价，可以为空 |

* 返回实例：

{

"status": true,

"info": null

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |

* 接口：setOneStudentResults
* 功能： 老师根据得分项评分
* 权限： 老师：可以查看所有学生的实验。
* API请求地址： 接口基本地址/v1/api/etOneStudentResults
* 请求方式 ： POST
* 请求实例：

{

"student\_id": "201710414201",

"status": true,

"info": null,

"data":

{

"grade\_id":"01",

"title":"实验一",

"request":48,

"format":18,

"expansion":15,

"comment":"本次实验做的不错"

}

}

* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| student\_id | 学号 |
| data | 实验的各得分项得分 |
| grade\_id | 成绩单编号 |
| comment | 本实验老师的评价，可以为空 |

* 返回实例：

{

"status": true,

"info": null

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |

## 4. 参照表

* [student](https://github.com/zemaochen/is_analysis/blob/master/test6/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#student)
* [grade](https://github.com/zemaochen/is_analysis/blob/master/test6/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#grade)
* [test](https://github.com/zemaochen/is_analysis/blob/master/test6/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1.md/#test)