

**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 | 2016级 |
| 学生姓名 | 郭高余 | 学号 | 201610414206 |
| 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** | | |  |

2019 年 5 月 1 日

目录

[一、 系统概述 3](#_Toc7970074)

[二、 系统总体结构 3](#_Toc7970075)

[三、 系统界面设计 4](#_Toc7970076)

[四、 系统用例图设计 9](#_Toc7970077)

[五、 类图设计 10](#_Toc7970078)

[六、 数据库设计 10](#_Toc7970079)

[1. User表（用户表） 10](#_Toc7970080)

[2. Teacher表（教师表） 11](#_Toc7970081)

[3. Student表（学生表） 12](#_Toc7970082)

[4. Course表（课程表） 12](#_Toc7970083)

[5. Score表（成绩表） 13](#_Toc7970084)

[6. Test表（实验项目表） 14](#_Toc7970085)

[七、 用例详细设计 15](#_Toc7970086)

[1. “查看学生列表”用例 15](#_Toc7970087)

[2. “成绩评定”用例 16](#_Toc7970088)

[3. “查看课程成绩”用例 18](#_Toc7970089)

[4. “修改密码”用例 19](#_Toc7970090)

[5. “修改用户信息”用例 20](#_Toc7970091)

[6. “查看用户信息”用例 21](#_Toc7970092)

[7. “登出”用例 22](#_Toc7970093)

[8. “登录”用例 23](#_Toc7970094)

[9. “添加学生”用例 24](#_Toc7970095)

[10. “添加课程”用例 25](#_Toc7970096)

[11. “添加课程评分细则”用例 26](#_Toc7970097)

[12. “查看课程列表”用例 27](#_Toc7970098)

[八、 接口详细设计 28](#_Toc7970099)

[1. addCourse 28](#_Toc7970100)

[2. addScoreItem 29](#_Toc7970101)

[3. addStudents 31](#_Toc7970102)

[4. getAllStu 32](#_Toc7970103)

[5. getSelfInfo 34](#_Toc7970104)

[6. getStudentScores 36](#_Toc7970105)

[7. login 38](#_Toc7970106)

[8. logout 39](#_Toc7970107)

[9. savePwd 41](#_Toc7970108)

[10. saveScore 42](#_Toc7970109)

[11. saveSelfInfo 44](#_Toc7970110)

[12. showCourseList 46](#_Toc7970111)

# 系统概述

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，二是查询自己的实验成绩。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：一是添加课程，二是添加每个实验的评分项，三是查看每个学生的成绩并且评分。

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。

系统平台只显示当学期的课程，不会显示非当学期的课程

在添加课程后导入学生，只能选择老师所在学校的学生，没有刻意建立班级、学校表，在实现时使用SQL的GROUP BY查询班级,以供老师选择，所谓SQL写得好少建两张表。

# 系统总体结构

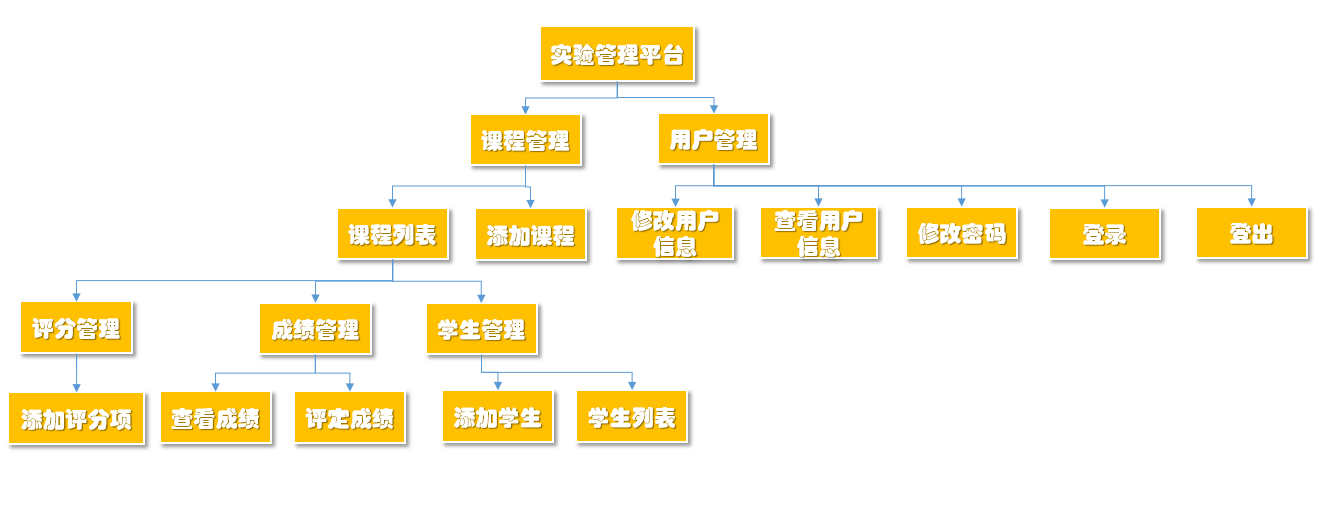


图 2-1 系统总体结构图

# 系统界面设计



图 3-1 登录界面

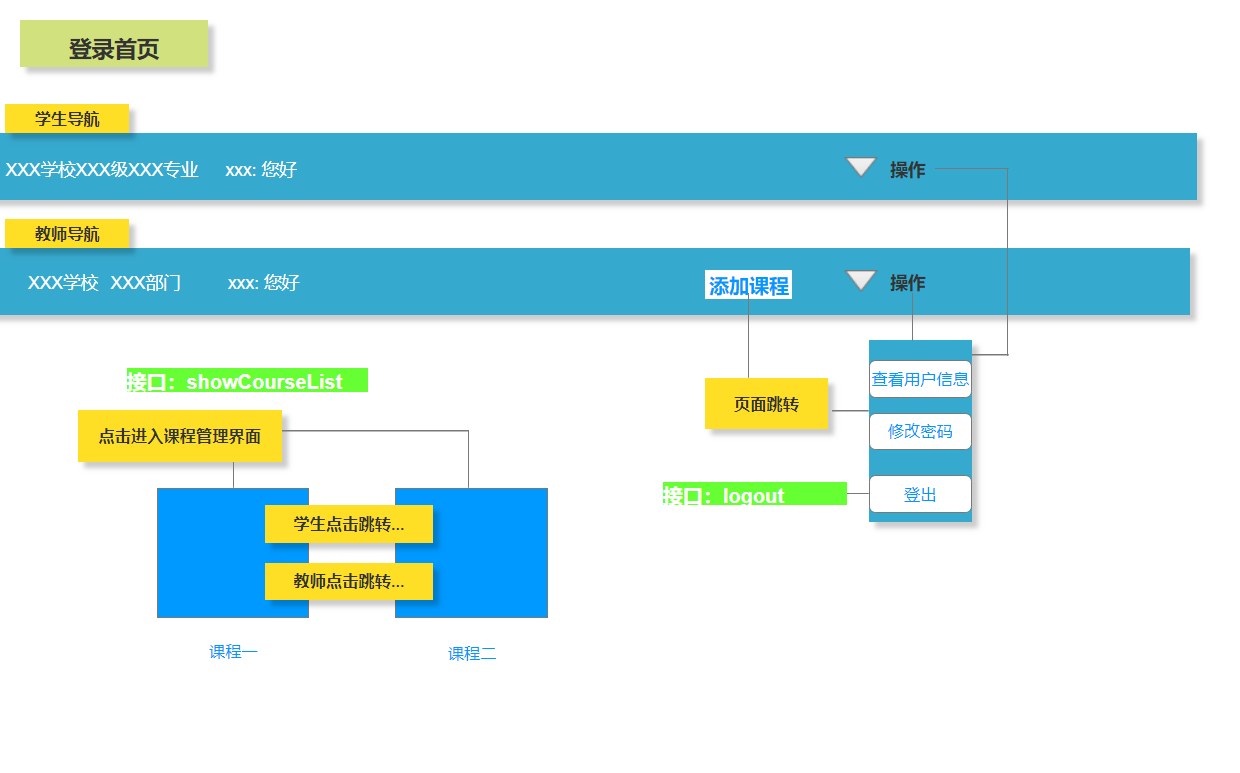


图 3-2 登录首页界面



图 3-3 用户信息界面



图 3-4 修改用户密码界面



图 3-5教师课程-学生管理界面



图 3-6 评定成绩及查看成绩界面



图 3-7 添加课程界面



图 3-8 添加课程评分细则界面



图 3-9 添加学生界面

# 系统用例图设计

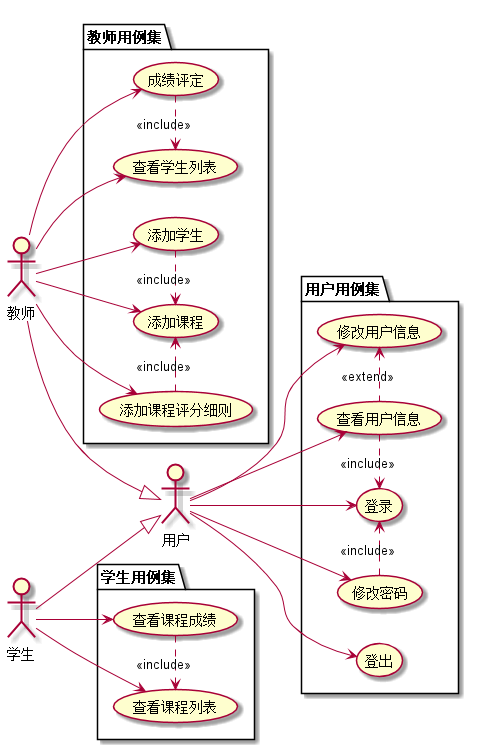


图 4-1 系统用例图

# 类图设计

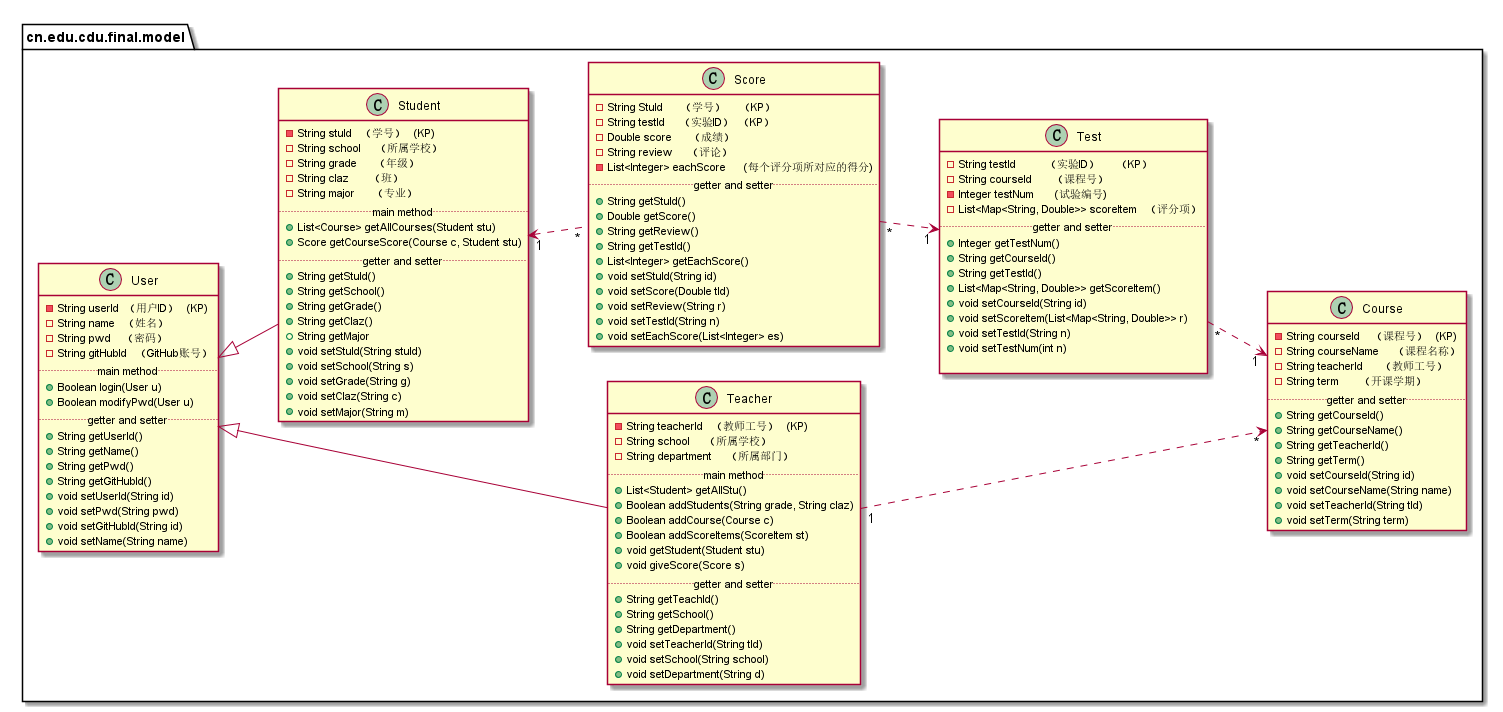


图 5-1 类图

# 数据库设计

1. User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 | NUMBER(8,0) | 主键 | 否 |  |  | 用户ID |
| NAME | VARCHAR2(50 BYTE) |  | 否 |  |  | 用户真实姓名 |
| GITHUB\_USERNAME | VARCHAR2(50 BYTE) |  | 是 | 空 |  | GitHub用户名 |
| PASSWORD | VARCHAR2(512 BYTE) |  | 是 | 空 |  | 加密存储密码，为空表示密码就是学号 |

1. Teacher表（教师表）

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| TEACHER\_ID | VARCHAR2(50 BYTE) | 主键 | 否 |  |  | 老师的编号 |
| USER\_ID | NUMBER(8,0) | 外键 | 是 |  |  | 老师的用户ID，USERS表的外键 |
| DEPARTMENT | VARCHAR2(400 BYTE) |  | 否 |  |  | 老师属于的部门 |
| SCHOOL | VARCHAR2(400 BYTE) |  | 否 |  |  | 老师所在的学校 |

1. 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 | VARCHAR2(50 BYTE) | 主键 | 否 |  |  | 学生的学号 |
| USER\_ID | NUMBER(8,0) | 外键 | 是 |  | 空 | 学生的用户ID，USERS表的外键，为空表示还没有建立用户 |
| MAJOR | VARCHAR2(20 BYTE) |  | 否 |  |  | 学生的专业 |
| CLAZ | VARCHAR2(20 BYTE) |  | 否 |  |  | 学生的班号 |
| GRADE | VARCHAR2(20 BYTE) |  | 否 |  |  | 学生的年级 |
| SCHOOL | VARCHAR2(400 BYTE) |  | 否 |  |  | 学生所属学校 |

1. 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 | VARCHAR2(50 BYTE) | 主键 | 否 |  |  | 课程号 |
| TEACHER\_ID | VARCHAR2(50 BYTE) | 外键 | 否 |  |  | 教师工号 |
| NAME | VARCHAR2(20 BYTE) |  | 否 |  |  | 课程名称 |
| TERM | VARCHAR2(20 BYTE) |  | 否 |  |  | 开课学期 |

1. Score表（成绩表）

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| TEST\_ID | VARCHAR2(50 BYTE) | 联合主键1，外键 | 否 |  |  | 实验ID，Test表外键 |
| STUDENT\_ID | VARCHAR2(50 BYTE) | 联合主键2，外键 | 否 |  |  | 学生的学号，Student表外键 |
| SCORE | NUMBER |  | 是 | 空 | 取值0-100 | 分数，这个值为空表示没有批改 |
| REVIEW | VARCHAR2(400 BYTE) |  | 是 | 空 |  | 老师对实验的评语 |
| EACH\_SCORE | VARCHAR2(50 BYTE) |  | 是 | 空 |  | 每个评分项分数组成的字符串，用分隔符分开 |

1. Test表（实验项目表）

| **字段** | **类型** | **主键，外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| TEST\_ID | VARCHAR2(50 BYTE) | 主键 | 否 |  |  | 实验ID |
| COURSE\_ID | VARCHAR2(50 BYTE) | 外键 | 否 |  |  | 课程ID |
| TEST\_NUM | NUMBER(6,0) |  | 否 |  |  | 实验编号 |
| SCORE\_ITEM | VARCHAR2(400 BYTE) |  | 是 | 空 |  | 实验的评分项，中间有分割符隔开各个评分项 |

# 用例详细设计

1. “查看学生列表”用例
2. **用例规约**

|  |
| --- |
|  |
| **用例名称** | **查看学生列表** |
| 功能 | 以表形式的显示出所有学生的信息 |
| 参与者 | 老师 |
| 前置条件 | 老师需要先登录 |
| 后置条件 |  |
| 主事件流 |  |
| 备选事件流 |  |

表 7-1 “查看学生列表”用例规约表

1. **业务流程**

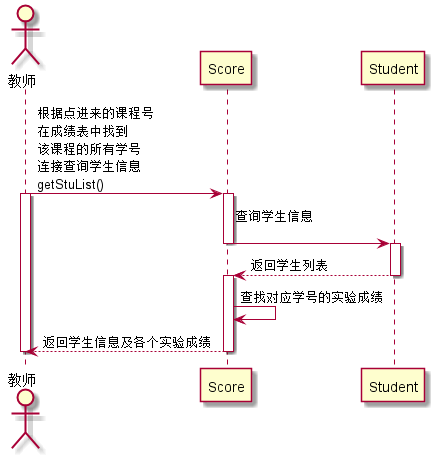
****

图 7-1 “查看学生列表”用例时序图

1. **界面设计**

界面参照：图 3-5

1. **API接口调用**

接口：getAllStu

1. **算法描述**
   1. 接口接收结果为一个映射列表（List<Map<~>>），分别装有学生信息和学生成绩。
   2. 成绩是来自Score表，其中有N次实验的成绩，若没有评定成绩则显示N。
   3. 平均成绩为各个实验成绩接收后运算出的结果。
2. **参照表**
3. Student表
4. Score表
5. Test表
6. “成绩评定”用例
7. **用例规约**

| **用例名称** | **成绩评定** |
| --- | --- |
| 功能 | 老师评定一个学生的实验成绩 |
| 参与者 | 老师 |
| 前置条件 | 查看成绩：评定成绩之前，老师需要先登录，并显示出一个学生的所有实验成绩和评语信息 |
| 后置条件 | 评定成绩提交之后，系统自动计算平均成绩 |
| 主事件流 | 1. 查看已有的成绩  2. 输入一个或者多个实验的成绩和评语  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 | 2a. 成绩必须是0至100之间，可以为空，但不能超界  1.提示用户重新输入成绩分数 |

表 7-2 “成绩评定”用例规约表

1. **业务流程**

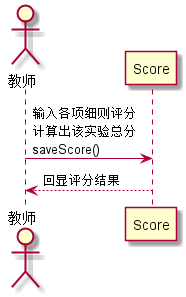
****

图 7-2 “成绩评定”用例时序图

1. **界面设计**

界面参照：图 3-6

1. **API接口调用**
   1. 接口1：getStudentScores
   2. 接口2：saveScore
2. **算法描述**

无

1. **参照表**
   1. Student表
   2. Score表
   3. Test表
   4. Course表
2. “查看课程成绩”用例
3. **用例规约**

|  |
| --- |
|  |
| **用例名称** | **查看课程成绩** |
| 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 |  |

表 7-3 “产看课程成绩”用例规约表

1. **业务流程**

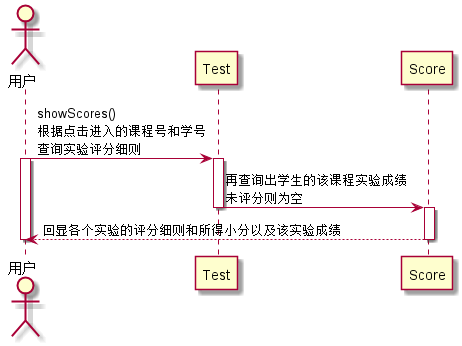
****

图 7-3 “查看课程成绩”用例时序图

1. **界面设计**

界面参照：图 3-6

1. **API接口调用**

接口：getStudentScores

1. **算法描述**

无

1. **参照表**
   1. Student表
   2. Score表
   3. Test表
2. “修改密码”用例
3. **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 | 必须先登录 |
| 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。 |

表 7-4 “修改密码”用例规约表

1. **业务流程**

无

1. **界面设计**

界面参照：图 3-4

1. **API接口调用**

接口1：savePwd

接口2：getSelfInfo

1. **算法描述**

无

1. **参照表**

User表

1. “修改用户信息”用例
2. **用例规约**

| **用例名称** | **修改用户信息** |
| --- | --- |
| 功能 | 修改用户个人信息 |
| 参与者 | 学生，老师 |
| 前置条件 | 必须先登录，并且查看用户的个人信息 |
| 后置条件 |  |
| 主事件流 | 1.用户填写用户信息  2.用户提交修改信息  3.系统存储修改后的用户信息 |
| 备选事件流 | 1a. 如果用户输入的信息为空  1.系统清空用户的该条信息 |

表 7-5 “修改用户信息”用例规约表

1. **业务流程**

无

1. **界面设计**

界面参照：图 3-3

1. **API接口调用**

接口1：getSelfInfo

接口2：saveSelfInfo

1. **算法描述**

无

1. **参照表**

User表

Student表

Teacher表

1. “查看用户信息”用例
2. **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 |  |

表 7-6 “查看用户信息”用例规约表

1. **业务流程**

无

1. **界面设计**

界面参照：图 3-3

1. **API接口调用**

接口：getSelfInfo

1. **算法描述**

无

1. **参照表**

Student表

Teacher表

User表

1. “登出”用例
2. **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 |  |
| 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（Cookie） |
| 备选事件流 | 1a. 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 |

表 7-7 “登出”用例规约表

1. **业务流程**

无

1. **界面设计**

界面参照：图 3-2

1. **API接口调用**

接口：logout

1. **算法描述**

无

1. **参照表**

User表

1. “登录”用例
2. **用例规约**

| **用例名称** | **登录** |
| --- | --- |
| 功能 | 登录平台 |
| 参与者 | 访客 |
| 前置条件 |  |
| 后置条件 | 登录成功后，跳转到主页 |
| 主事件流 | 1. 访客输入用户名和密码，选择用户类型 2.系统判断用户名，密码，用户类正确，允许登录 3.系统在客户端以Cookie形式存储登录用户信息，保持登录的持久性。 |
| 备选事件流 | 1a. 输入的用户名或者密码为空  1.提示重新输入  2.访客重新提交登录信息  2a.系统判断用户名，密码，用户类不正确，不允许登录  1.提示重新输入  2.访客重新提交登录信息 |

表 7-8 “登录”用例规约表

1. **业务流程**

无

1. **界面设计**

界面参照：图 3-1

1. **API接口调用**

接口：login

1. **算法描述**

无

1. **参照表**

User表

1. “添加学生”用例
2. **用例规约**

| **用例名称** | **添加学生** |
| --- | --- |
| 功能 | 老师添加授课学生 |
| 参与者 | 老师 |
| 前置条件 | 添加授课学生之前，老师需要先登录并且添加课程以及添加课程实验评分细则 |
| 后置条件 | 授课学生班级提交之后，系统自动保存学生名单，并生成对应的实验成绩表 |
| 主事件流 | 1.选择与自己相同学校的学生班级并提交 |
| 备选事件流 | 1a. 学生班级不能为空  1.提示用户重新选择 |

表 7-9 “添加学生”用例规约表

1. **业务流程**

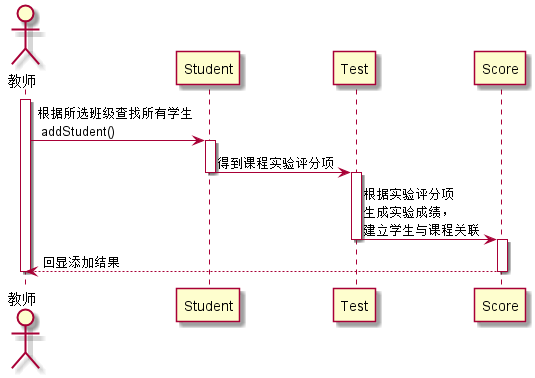
****

图 7-9 “添加学生”用例时序图

1. **界面设计**

界面参照：图 3-9

1. **API接口调用**

接口：addStudents

1. **算法描述**

无

1. **参照表**

Student表

Test表

1. “添加课程”用例
2. **用例规约**

| **用例名称** | **添加课程** |
| --- | --- |
| 功能 | 老师开设课程 |
| 参与者 | 老师 |
| 前置条件 | 添加课程之前，老师需要先登录 |
| 后置条件 | 课程信息提交之后，系统自动保存课程信息，继续添加课程实验评分项 |
| 主事件流 | 1.输入课程信息并提交 2.跳转添加实验评分项界面 |
| 备选事件流 | 1a. 课程信息每项不能为空  1.提示用户重新输入 |

表 7-10 “添加课程”用例规约表

1. **业务流程**

无

1. **界面设计**

界面参照：图 3-7

1. **API接口调用**

接口1：getSelfInfo

接口2：addCourse

1. **算法描述**

无

1. **参照表**

User表

Teacher表

Course表

1. “添加课程评分细则”用例
2. **用例规约**

| **用例名称** | **添加课程评分细则** |
| --- | --- |
| 功能 | 老师添加实验评分项 |
| 参与者 | 老师 |
| 前置条件 | 添加课程评分细则之前，老师需要先登录并且添加课程 |
| 后置条件 | 评分项提交之后，系统自动保存评分项 |
| 主事件流 | 1.输入课程评分细则信息并提交 |
| 备选事件流 | 1a. 课程评分细则信息每项不能为空  1.提示用户重新输入 |

表 7-11 “添加课程评分细则”用例规约表

1. **业务流程**

无

1. **界面设计**

界面参照：图 3-8

1. **API接口调用**

接口：addScoreItem

1. **算法描述**

无

1. **参照表**

Course表

Test表

1. “查看课程列表”用例
2. **用例规约**

|  |
| --- |
|  |
| **用例名称** | **查看课程列表** |
| 功能 | 以表形式的显示出所有开设课程的信息 |
| 参与者 | 学生、老师 |
| 前置条件 | 用户需要先登录，并且老师已经开设了课程 |
| 后置条件 | 显示所开设的课程 |
| 主事件流 |  |
| 备选事件流 |  |

表 7-12 “查看课程列表”用例规约表

1. **业务流程**

无

1. **界面设计**

界面参照：图 3-2

1. **API接口调用**

接口：showCourseList

1. **算法描述**

无

1. **参照表**

User表

Student表

Teacher表

Course表

# 接口详细设计

1. addCourse
2. **用例**

添加课程

1. **权限**

老师：登录后可以添加课程

1. **功能**

开设课程

1. **API请求地址**

接口基本地址/v1/api/addCourse/

1. **请求方式**

POST

1. **请求实例**

|  |
| --- |
| {  "teacher\_id":"10001",  "name":"IS",  "term":"2019-3",  } |

1. **请求参数说明**

| **参数名称** | **说明** |
| --- | --- |
| teacher\_id | 教师工号 |
| name | 课程名称 |
| term | 开课学期，由系统自动根据时间生成 |

表 8-1-1 请求参数说明表

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null,  "course\_id":"1000001"    } |

1. **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 |
| course\_id | 返回系统自增生成的课程ID用于下一步添加评分细则 |

表 8-1-2 请求返回参数说明表

1. addScoreItem
2. **用例**

添加课程评分细则

1. **权限**

老师：登录后添加课程后必须添加课程实验评分细则

1. **功能**

添加课程实验评分细则

1. **API请求地址**

接口基本地址/v1/api/addScoreItems/

1. **请求方式**

POST

1. **请求实例**

|  |
| --- |
| {  "course\_id":"10000001",  "item":[  {    "test\_n":1  "实验完整":30,  "有改进":30,  “细节完整”:40  },  {  ...其他实验的评分细则  }  ]  } |

1. **请求参数说明**

| **参数名称** | **说明** |
| --- | --- |
| course\_id | 上一页面添加课程后生成的课程ID |
| item | 实验评分细则映射列表 |

表 8-2-1 请求参数说明表

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null,  } |

1. **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 |

表 8-2-2 返回参数说明表

1. addStudents
2. **用例**

添加学生

1. **权限**

老师：登录后并且已经添加了课程和课程评分细则后才能添加学生

1. **功能**

添加授课学生

1. **API请求地址**

接口基本地址/v1/api/addStudents/

1. **请求方式**

 POST

1. **请求实例**

|  |
| --- |
| {  "course\_id":"10000001",  "stu":[  {  "school":"成都大学",  "年级":"2016",  "班":2,  "专业":"软件工程"  },  {  ...其他班级  }  ]  } |

1. **请求参数说明**

| **参数名称** | **说明** |
| --- | --- |
| course\_id | 添加课程时生成的课程号 |
| stu | 班级列表，根据学校、班级、专业从Student导入学生，生成对应的成绩表 |

表 8-3-1 请求参数说明表

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null,  } |

1. **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 |

表 8-3-2 返回参数说明表

1. getAllStu
2. **用例**

查看学生列表

1. **权限**

老师：可以看到所授课学生信息和成绩

1. **功能**

返回所有授课学生的列表

1. **API请求地址**

接口基本地址/v1/api/getAllStu/<course\_id>

1. **请求方式**

GET

1. **请求参数说明**

course\_id:课程的ID号

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null,  "total": 121,  "data": [  {  “stu\_info”:{  "github\_username":"ApllauseWow"  "stu\_id":"201610414206",  "name":"HolyKowk",  "claz":"2",  ...  },  "stu\_score":{  "test\_1":"90",  "test\_2":"85",  "test\_3":"88",  ...  }  },  {  ...其他学生  }  ]  } |

1. **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 |
| total | 返回学生人数 |
| data | 所有学生的信息成绩映射关系数组 |
| stu\_info | 学生基本信息列表 |
| GITHUB\_USERNAME | GITHUB 用户名 |
| STUDENT\_ID | 学号 |
| CLAZ | 班级 |
| NAME | 真实姓名 |
| stu\_score | 学生成绩列表 |
| test\_1 | 第一次试验成绩 |
| test\_2 | 第二次实验成绩 |
| test\_3 | 第三次实验成绩 |

表 8-4-1 返回参数说明表

1. getSelfInfo
2. **用例**

查看用户信息,修改用户信息

1. **权限**

学生/老师：查看自己的信息，必须先登录

1. **功能**

查看用户的所有信息

1. **API请求地址**

接口基本地址/v1/api/getSelfInfo/<user\_id>

1. **请求方式**

GET

1. **请求参数说明**

user\_id: 用户的ID号。对应表USERS.USER\_ID的值

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null,  "ID":"201610414206",  "name":"HolyKowk",  "claz":"2"  "github\_username":"ApplauseWow",  "type":"学生" ,  ...  } |

1. **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 |
| ID | 学号或者工号 |
| name | 用户的真实姓名 |
| claz | 所属班 |
| github\_username | gitHub用户名 |
| type | 用户类型：老师或者学生 |

表 8-5-1 返回参数说明表

1. getStudentScores
2. **用例**

查看课程成绩，成绩评定

1. **权限**

学生：只能查看自己的该课程成绩，即接口参数student\_id必须等于登录学生的student\_id，course\_id等于点击进入的course\_id

老师：可以查看该学生的成绩以及对成绩评定

1. **功能**

返回一个学生的所有实验成绩和实验评价以及该实验的评分细则

1. **API请求地址**

接口基本地址/v1/api/getStudentScores/<student\_id>&<course\_id>

1. **请求方式**

GET

1. **请求参数说明**

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

表 8-6-1 请求参数说明表

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null,  "student\_id": "201610414206",  "github\_username": "ApplauseWow",  "class": "软件(本)16-2",  "name": "HolyKwok",  "total": 6,  "avgresult":90.5,  "data": [  {  "test\_id":1001,  "test\_n":1  "score": 91,  "review":"本实验做得好",  "item":{  "实验完整":30,  "用例完整":30,  "有改进":40  },  "each\_score":{  "1":30,  "2":30,  "3":31  }  },  {  ...其他实验  }  ]  } |

1. **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 |
| student\_id | 学号 |
| github\_username | 学生的gitHub用户名 |
| class | 班级 |
| name | 真实姓名 |
| total | 实验总数 |
| data | 所有实验的成绩和评语 |
| test\_id | 实验ID |
| test\_n | 实验编号 |
| score | 本实验成绩，可以为空，为空表示没有批改，没有批改的实验不入平均成绩，为0分则要计入平均成绩，所以成绩为空和为0是不一样的。 |
| review | 本实验老师的评价，可以为空 |
| item | 本实验各个评分细则 |
| each\_score | 各个评分细则得分 |

表 8-6-2 返回参数说明表

1. login
2. **用例**

登录

1. **权限**

访客

1. **功能**

登录到平台

1. **API请求地址**

 接口基本地址/v1/api/login

1. **请求方式**

POST

1. **请求实例**

|  |
| --- |
| {  "id":"201610414206",  "password":"201610414206",  "type":"学生"  } |

1. **请求参数说明**

| **参数名称** | **说明** |
| --- | --- |
| id | 学生学号或者老师的工号，由type决定。 |
| password | 用户的密码 |
| type | 用户类型，学生或者老师 |

表 8-7-1 请求参数说明表

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null,  } |

1. **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 |

表 8-7-2 返回参数说明表

1. logout
2. **用例**

登出

1. **权限**

学生/老师已经登录

1. **功能**

已经登录的用户登出平台

1. **API请求地址**

接口基本地址/v1/api/logout

1. **请求方式**

POST

1. **请求实例**

|  |
| --- |
| {  "user\_id":201610414206  } |

1. **请求参数说明**

user\_id: 已经登录用户的user\_id值，对应表USERS.USER\_ID的值

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null,  } |

1. **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 |

表 8-8-1 返回参数说明表

1. savePwd
2. **用例**

修改密码

1. **权限**

学生/老师：修改自己的密码，必须先登录

1. **功能**

修改（设置）用户的密码

1. **API请求地址**

接口基本地址/v1/api/savePwd

1. **请求方式**

POST

1. **请求实例**

|  |
| --- |
| {  "user\_id":10001,  "password":"201610414206"  } |

1. **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 |
| password | 用户的密码。不能为空 |

表 8-9-1 请求参数说明表

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null,  } |

1. **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 |

表 8-9-2 返回参数说明表

1. saveScore
2. **用例**

评定成绩

1. **权限**

老师：可以查看学生的所有实验成绩

1. **功能**

设置一个学生的部分实验成绩和评语。

输入参数each\_score不为空。

输入参数each\_score为空，表示未批改。

自动根据each\_score所打分数生成实验总成绩score。

1. **API请求地址**

接口基本地址/v1/api/saveScore/

1. **请求方式**

POST

1. **请求实例**

|  |
| --- |
| {  "student\_id": "201610414206",  "total": 6,  "data": {  {  "test\_id":10001,  "test\_n":1  "score": 91,  "review":"本实验做得好",  “each\_score”:{  "1":30,  "2":30,  "3":31  }  },  {  ...其他实验  }  }  } |

1. **请求参数说明**

| **参数名称** | **说明** |
| --- | --- |
| student\_id | 学号 |
| total | 本次批改的所有实验的总数，小于等于全部实验的总数 |
| data | 实验的成绩和评语 |
| test\_id | 实验ID |
| test\_n | 实验编号 |
| score | 本实验成绩，可以为空，为空表示没有批改。 |
| review | 本实验老师的评价，可以为空 |
| each\_score | 每个评分项所得分数 |

表 8-10-1 请求参数说明表

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null  } |

1. **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 |

表 8-10-2 返回参数说明表

1. saveSelfInfo
2. **用例**

修改用户信息

1. **权限**

学生/老师：必须先登录

1. **功能**

修改用户的个人信息

1. **API请求地址**

接口基本地址/v1/api/saveSelfInfo

1. **请求方式**

POST

1. **请求实例**

|  |
| --- |
| {  "id":"201610414206",  "github\_username":"ApplauseWow",  "name":"HolyKwok",  ...  } |

1. **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 |
| github\_username | 用户GitHub用户名。 |
| name | 用户姓名 |
| ... | ... |

表 8-11-1 请求参数说明表

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null  } |

1. **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 |

表 8-11-2 返回参数说明表

1. showCourseList
2. **用例**

查看课程列表

1. **权限**

学生、老师：登录后可以查看所开设的课程

1. **功能**

返回所有开设课程的列表

1. **API请求地址**

接口基本地址/v1/api/showCourseList/

1. **请求方式**

GET

1. **请求实例**

|  |
| --- |
| {  "stu\_id":"201610414206"  } |

1. **请求参数说明**

stu\_id/teacher\_id:学生学号/教师工号

1. **返回实例**

|  |
| --- |
| {  "status": true,  "info": null,  "total": 4,  "data": [  {  "course\_id":100001,  "name":"IS",  "teacher\_id":"10001",  "term":“2019-3”  },  {  ...其他课程  }  ]    } |

1. **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 |
| total | 返回课程数 |
| data | 所有课程的信息数组 |
| course\_id | 课程ID |
| teacher\_id | 开课教师工号 |
| name | 课程名称 |
| term | 开课学期 |

表 8-12-1 返回参数说明表