

**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 | 肖清宇 | 学号 | 201610414422 |
| 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

[1.概述 4](#_Toc8027280)

[2.系统总体结构 4](#_Toc8027281)

[3.用例图设计 5](#_Toc8027282)

[4.类图设计 6](#_Toc8027283)

[5.数据库设计 6](#_Toc8027284)

[5-1 用户表 USERS 6](#_Toc8027285)

[5-2 用户状态表 USER\_STATUS 7](#_Toc8027286)

[5-3 教师表 TEACHERS 8](#_Toc8027287)

[5-4 部门表 DEPARTMENTS 8](#_Toc8027288)

[5-5 学生表 STUDENTS 9](#_Toc8027289)

[5-6 专业表 MAJORS 9](#_Toc8027290)

[5-7 课程表 COURSES 9](#_Toc8027291)

[5-8 学期表 TERMS 10](#_Toc8027292)

[5-9 开课课程教师关联表 COURSE\_TEACHER 10](#_Toc8027293)

[5-10 课程专业学期关联表 COURSE\_MAJOR\_TERM 11](#_Toc8027294)

[5-11 开课课程学生关联表 COURSE\_STUDENTS 11](#_Toc8027295)

[5-12 实验表 TESTS 12](#_Toc8027296)

[5-13 评分标准表 STANDARDS 12](#_Toc8027297)

[5-14 详细评分表 DETAIL\_SCORES 13](#_Toc8027298)

[5-15 实验总评分表 TEST\_TOTAL\_SCORES 13](#_Toc8027299)

[6.用例及界面详细设计 14](#_Toc8027300)

[6-1“登录”用例 14](#_Toc8027301)

[6-2 “登出”用例 16](#_Toc8027302)

[6-3“修改密码”用例 18](#_Toc8027303)

[6-4 “修改用户信息”用例 19](#_Toc8027304)

[6-5“查看用户信息”用例 21](#_Toc8027305)

[6-6 “选择课程”用例 23](#_Toc8027306)

[6-7“查看课程”用例 25](#_Toc8027307)

[6-8 “查看学生信息”用例 27](#_Toc8027308)

[6-9“查看实验”用例 29](#_Toc8027309)

[6-10“发布实验”用例 31](#_Toc8027310)

[6-11 “查看成绩”用例 33](#_Toc8027311)

[6-12“评定成绩”用例 34](#_Toc8027312)

[7.接口描述 36](#_Toc8027313)

[7-1 login 36](#_Toc8027314)

[7-2 loginout 37](#_Toc8027315)

[7-3 setPassword 38](#_Toc8027316)

[7-4 updateUserInfo 39](#_Toc8027317)

[7-5 getUserInfo 40](#_Toc8027318)

[7-6 getOpenCourseByTeacher 41](#_Toc8027319)

[7-7 getOpenCourseByStudent 42](#_Toc8027320)

[7-8 setTeacherSelectCourse 43](#_Toc8027321)

[7-9 setStudentSelectCourse 44](#_Toc8027322)

[7-10 getCourseByTeacher 45](#_Toc8027323)

[7-11 getCourseByStudent 46](#_Toc8027324)

[7-12 getCourseTestAllStudents 47](#_Toc8027325)

[7-13 getCourseAllTests 48](#_Toc8027326)

[7-14 setNewCourseTest 49](#_Toc8027327)

[7-15 getOneStudentResults 50](#_Toc8027328)

[7-16 setOneStudentResults 53](#_Toc8027329)

## 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，二是查询自己的实验成绩,三是选择课程，四是查看实验。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：一是批改每个学生的成绩，二是查看每个学生的成绩，三是选择课程。四是发布实验。
* 系统基于三多设置，多课程、多学期、多评分项。
* 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。
* 老师可以对每个发布实验自定义评分标准，评分标准动态设定。
* 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

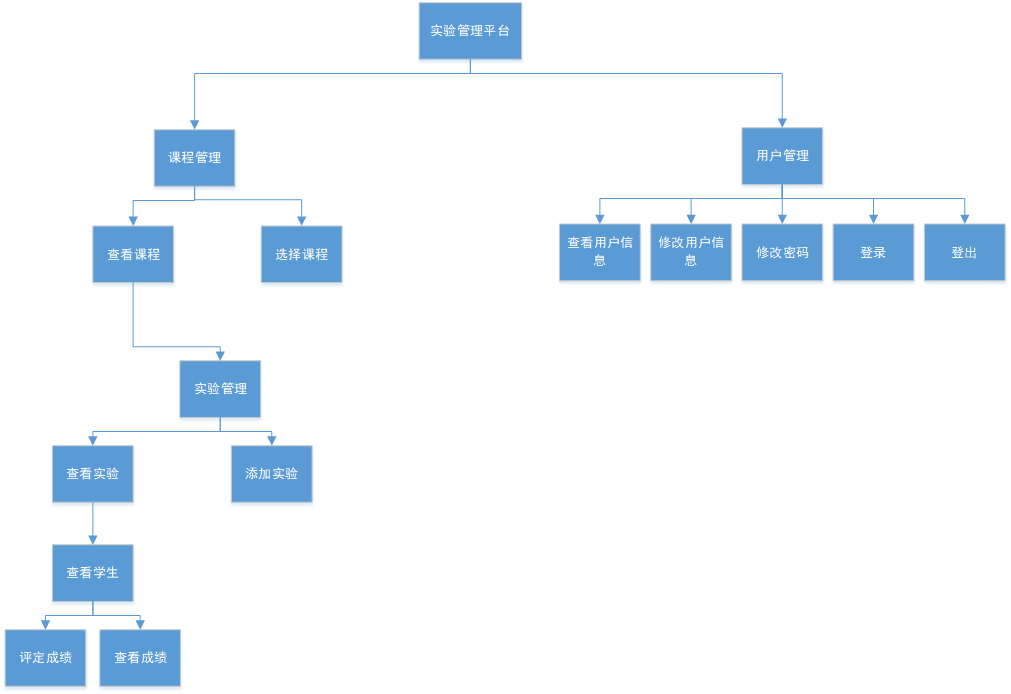


图2-1系统总体概述

## 3.用例图设计

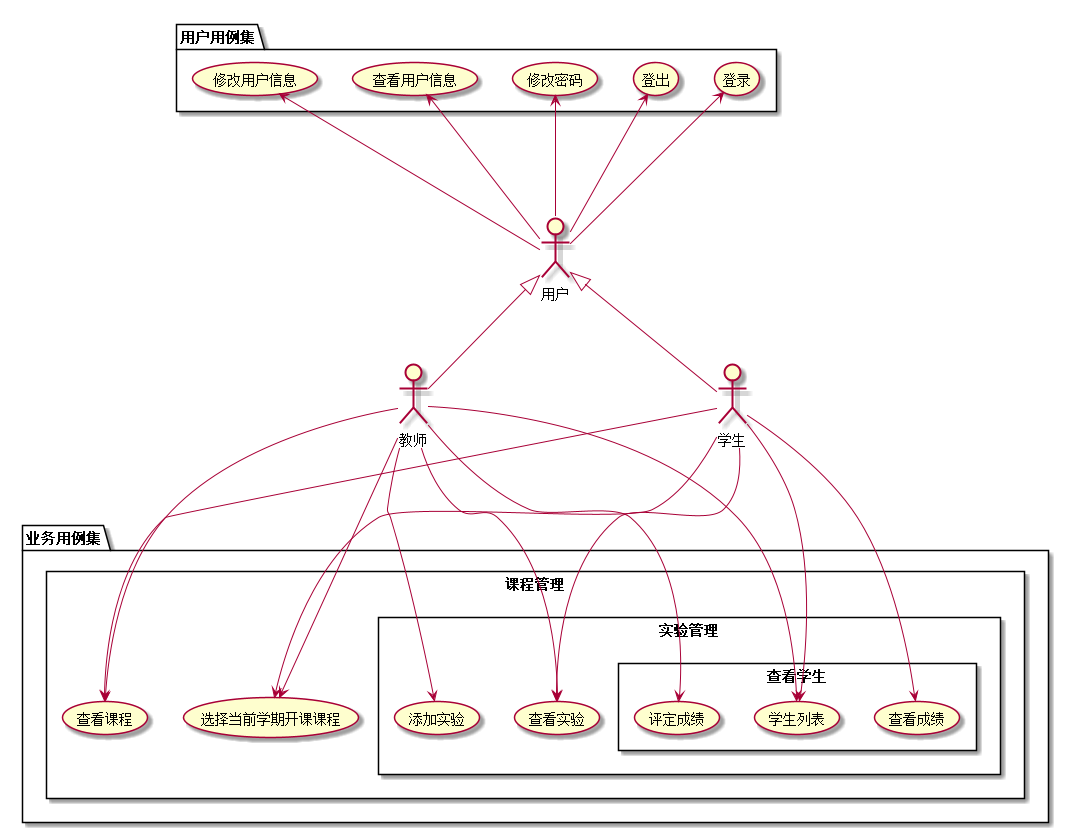


图3-1用例图设计

## 4.类图设计

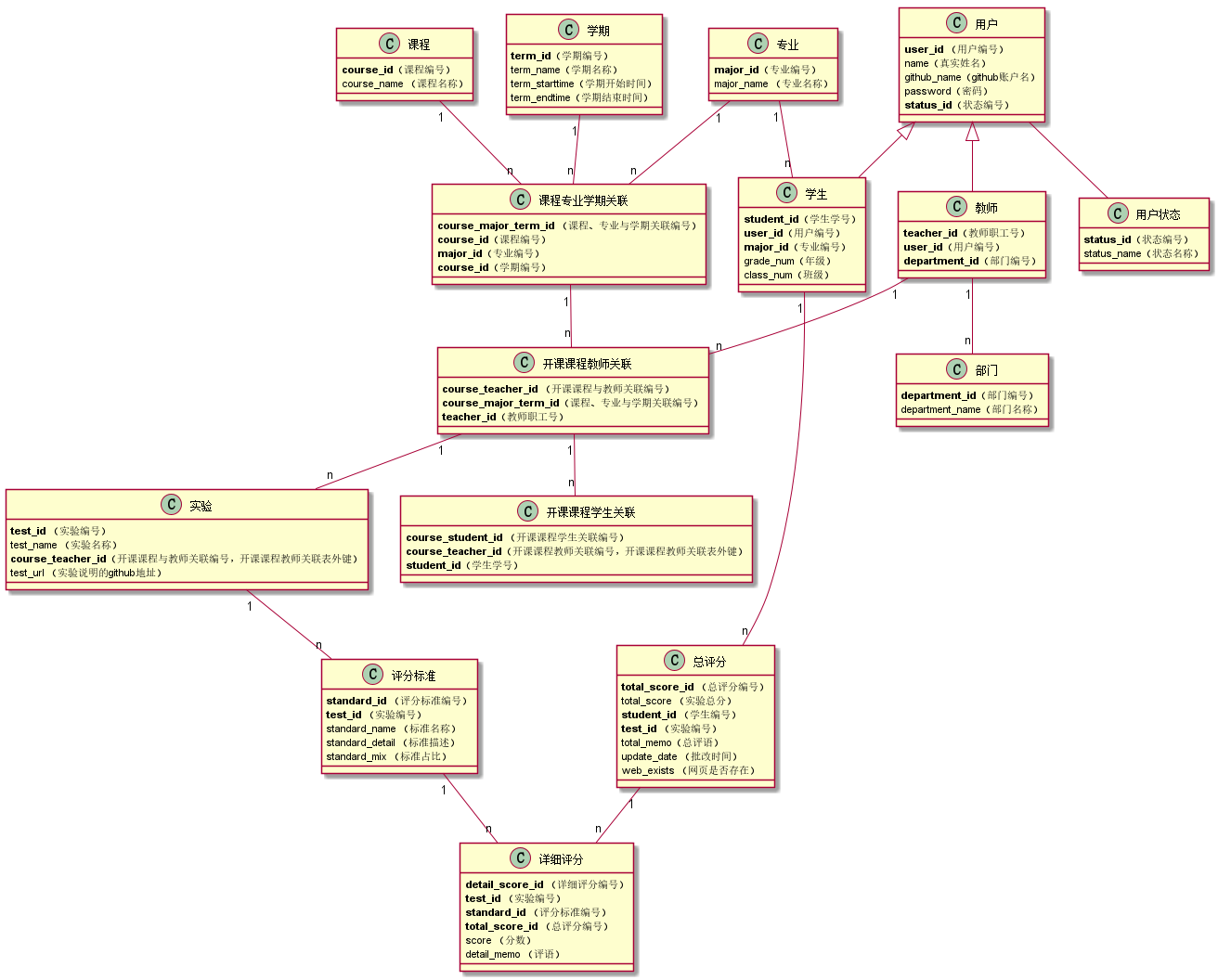


图4-1类图

## 5.数据库设计

### 5-1 用户表 USERS

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| user\_id | NUMBER(10,0) | 主键 | 否 |  |  | 用户id |
| name | VARCHAR2(128BYTE) |  | 否 |  |  | 真实姓名 |
| github\_name | VARCHAR2(128BYTE) |  | 是 | 空 |  | 用户github账户名 |
| update\_date | DATE |  | 是 | 空 |  | github用户名修改时间 |
| password | VARCHAR(512BYTE) |  | 是 | 空 |  | 用户密码 |
| status\_id | Number(1,0) | 外键 | 否 |  |  | 用户状态，用户状态表外键 |

### 5-2 用户状态表 USER\_STATUS

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| status\_id | NUMBER(10,0) | 主键 | 否 |  |  | 状态id |
| status\_name | VARCHAR2(64BYTE) |  | 否 |  |  | 状态名称 |

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| teacher\_id | NUMBER(10,0) | 主键 | 否 |  |  | 教师id，为教师入职时职工号 |
| user\_id | NUMBER(10,0) | 外键 | 否 |  |  | 用户id，用户表外键 |
| department\_id | VARCHAR2(400 BYTE) | 外键 | 否 |  |  | 教师所属部门，部门表外键 |

### 5-3 教师表 TEACHERS

### 5-4 部门表 DEPARTMENTS

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| department\_id | NUMBER(10,0) | 主键 | 否 |  |  | 教师id，为教师入职时职工号 |
| department\_name | VARCHAR2(128BYTE) |  | 否 |  |  | 部门名称 |

### 5-5 学生表 STUDENTS

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| student\_id | NUMBER(12,0) | 主键 | 否 |  |  | 学生id，学生学号 |
| user\_id | NUMBER(10,0) | 外键 | 否 |  |  | 用户id，用户表外键 |
| major\_id | NUMBER(10,0) | 外键 | 否 |  |  | 学生所属专业，专业表外键 |
| grade\_num | number(4,0) |  | 否 |  |  | 学生年级 |
| class\_num | number(4,0) |  | 否 |  |  | 学生班级 |

### 5-6 专业表 MAJORS

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| major\_id | NUMBER(10,0) | 主键 | 否 |  |  | 专业id |
| major\_name | VARCHAR2(128BYTE) |  | 否 |  |  | 专业名称 |

### 5-7 课程表 COURSES

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| course\_id | NUMBER(12,0) | 主键 | 否 |  |  | 课程id |
| course\_name | VARCHAR2(512BYTE) |  | 否 |  |  | 课程名称 |

### 5-8 学期表 TERMS

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| term\_id | NUMBER(6,0) | 主键 | 否 |  |  | 学期id |
| term\_name | VARCHAR2(64BYTE) |  | 否 |  |  | 学期名称 |
| term\_starttime | DATE |  | 否 |  |  | 学期开始时间 |
| term\_endtime | DATE |  | 否 |  |  | 学期结束时间 |

### 5-9 开课课程教师关联表 COURSE\_TEACHER

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| course\_teacher\_id | NUMBER(12,0) | 主键 | 否 |  |  | 开课课程教师关联id |
| course\_major\_term\_id | NUMBER(12,0) | 外键 | 否 |  |  | 课程、专业与学期关联编号，课程专业学期关联表外键 |
| teacher\_id | NUMBER(10,0) | 外键 | 否 |  |  | 教师id |

### 5-10 课程专业学期关联表 COURSE\_MAJOR\_TERM

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| course\_major\_term\_id | NUMBER(12,0) | 主键 | 否 |  |  | 课程、专业与学期关联编号 |
| course\_id | NUMBER(12,0) | 外键 | 否 |  |  | 课程id，课程表外键 |
| major\_id | NUMBER(10,0) | 外键 | 否 |  |  | 专业id，专业表外键 |
| term\_id | Number(6,0) | 外键 | 否 |  |  | 开课学期id，学期表外键 |

### 5-11 开课课程学生关联表 COURSE\_STUDENTS

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| course\_student\_id | NUMBER(12,0) | 主键 | 否 |  |  | 开课课程与学生关联编号 |
| course\_teacher\_id | NUMBER(12,0) | 外键 | 否 |  |  | 开课课程教师关联id，开课课程教师关联表外键 |
| student\_id | NUMBER(12,0) | 外键 | 否 |  |  | 学生id，学生表外键 |

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| test\_id | NUMBER(6,0) | 主键 | 否 |  |  | 实验id |
| test\_name | VARCHAR2(256BYTE) | 否 |  |  | 实验名称 |  |
| course\_teacher\_id | NUMBER(12,0) | 外键 | 否 |  |  | 开课课程教师关联id，开课课程教师关联表外键 |
| test\_url | VARCHAR(256BYTE) |  | 否 |  |  | 实验说明github地址 |

### 5-12 实验表 TESTS

### 5-13 评分标准表 STANDARDS

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| standard\_id | NUMBER(6,0) | 主键 | 否 |  |  | 评分标准id |
| test\_id | NUMBER(6,0) | 外键 | 否 |  |  | 实验id |
| standard\_name | VARCHAR2(128BYTE) |  | 否 |  |  | 评分标准名称 |
| standard\_detail | VARCHAR2(512BYTE) |  | 否 |  |  | 评分标准描述 |
| standard\_mix | NUMBER(2,0) |  | 否 |  |  | 评分标准占总分比例 |

### 5-14 详细评分表 DETAIL\_SCORES

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| detail\_score\_id | NUMBER(6,0) | 主键 | 否 |  |  | 详细评分id |
| test\_id | NUMBER(6,0) | 外键 | 否 |  |  | 实验id |
| standard\_id | NUMBER(6,0) | 外键 | 否 |  |  | 评分标准id |
| total\_score\_id | NUMBER(6,0) | 外键 | 否 |  |  | 总评分id，实验总分表外键 |
| score | NUMBER |  | 是 | 空 | 0-100 | 详细评分分数，分数为空为未批改 |
| detail\_memo | VARCHAR2(400BYTE) |  | 是 | 空 |  | 评语 |

### 5-15 实验总评分表 TEST\_TOTAL\_SCORES

| **字段** | **类型** | **主键、外键** | **可以为空** | **默认值** | **约束** | **说明** |
| --- | --- | --- | --- | --- | --- | --- |
| total\_score\_id | NUMBER(6,0) | 主键 | 否 |  |  | 总评分id |
| total\_score | NUMBER |  | 是 | 空 | 0-100 | 总评分分数，分数为空为未批改 |
| test\_id | NUMBER(6,0) | 外键 | 否 |  |  | 实验id |
| student\_id | NUMBER(12,0) | 外键 | 否 |  |  | 学生id，学生表外键 |
| total\_memo | VARCHAR2(400BYTE) |  | 是 | 空 |  | 总评语 |
| update\_date | DATE |  | 是 | 空 |  | 老师批改时间，为空为未批改 |
| web\_exists | BOOLEAN |  | 是 | 空 |  | 实验网页是否存在，空为不存在 |

## 6.用例及界面详细设计

### 6-1“登录”用例

#### 1. 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，用户角色类不正确，不允许登录     1.提示重新输入     2.访客重新提交登录信息 |

#### 2. 业务流程

无

#### 3. 界面设计

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/登录.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E7%99%BB%E5%BD%95.html)
* API接口调用
  + 接口1：[login](#_7-1_login_1)

#### 4. 算法描述

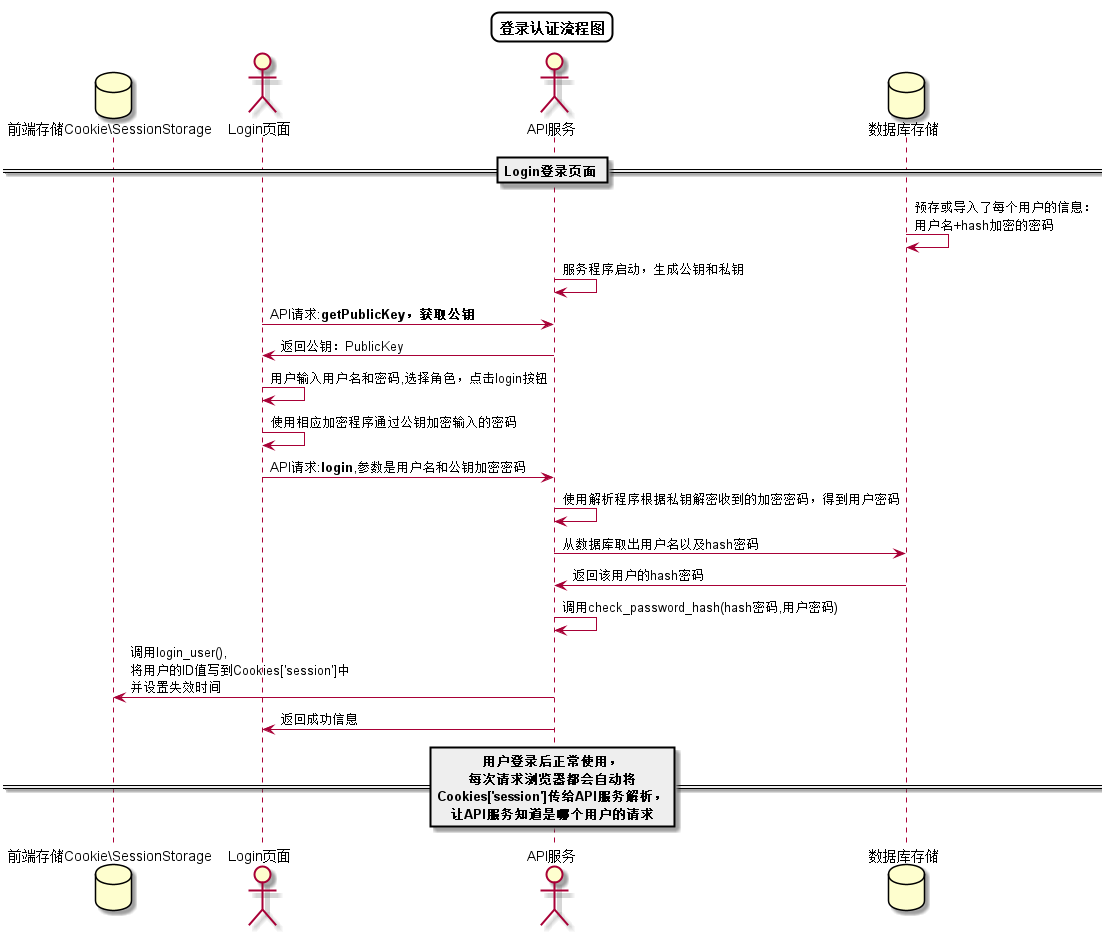


图 6-1

#### 5. 参照表

* [USERS](#_5-1.用户表_USERS)

### 6-2 “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 |  |
| 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（Cookie |

#### 2. 业务流程

无

#### 3. 界面设计

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/登出\_顶部菜单.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E7%99%BB%E5%87%BA_%E9%A1%B6%E9%83%A8%E8%8F%9C%E5%8D%95.html)
* API接口调用
  + 接口1：[loginout](#_7-2_loginout)

#### 4. 算法描述

无

#### 5. 参照表

* [USERS](#_1.用户表_USERS)

### 6-3“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 | 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.用户填写旧密码  2.用户填写新密码（两次输入）  3.用户提交修改信息  4.系统存储修改后密码 |
| 备选事件流 | 1a.用户旧密码与数据库存储密码不符    1.系统提示旧密码不正确     2. 用户重新填写并提交。 2a. 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

无

#### 3. 界面设计

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/修改密码.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E4%BF%AE%E6%94%B9%E5%AF%86%E7%A0%81.html)
* API接口调用
  + 接口1：[setPassword](#_7-3_setPassword)

#### 4. 算法描述

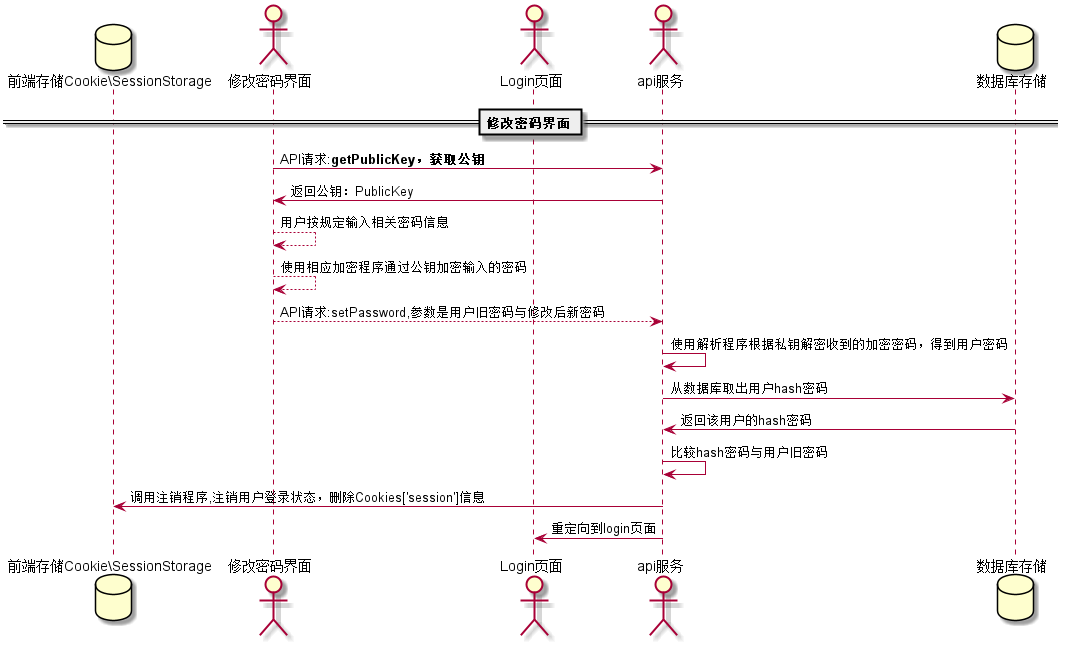


图 6-2

#### 5. 参照表

* [USERS](#_5-1.用户表_USERS)

### 6-4 “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 | 必须先登录，并且查看用户现有的GitHub用户名 |
| 后置条件 |  |
| 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.提示重新输入 |

#### 2. 业务流程

无

#### 3. 界面设计

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/修改用户信息.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E4%BF%AE%E6%94%B9%E7%94%A8%E6%88%B7%E4%BF%A1%E6%81%AF.html)
* API接口调用
  + 接口1：[getUserInfo](#_7-5_getUserInfo)
  + 接口2：[updateUserInfo](#_7-4_updateUserInfo)

#### 4. 算法描述

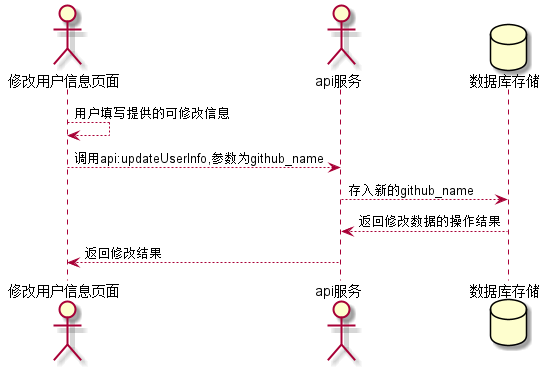


图 6-3

#### 5. 参照表

* [USERS](#_5-1.用户表_USERS)

### 6-5“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

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/查看用户信息.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E6%9F%A5%E7%9C%8B%E7%94%A8%E6%88%B7%E4%BF%A1%E6%81%AF.html)
* API接口调用
  + 接口1：[getUserInfo](#_7-5_getUserInfo)

#### 4. 算法描述

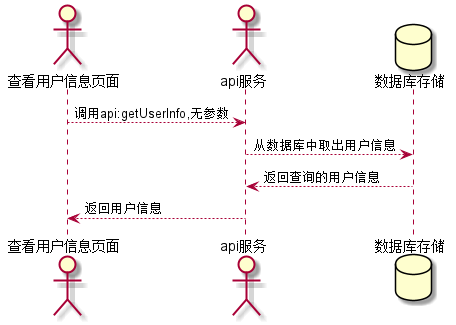


图 6-4

#### 5. 参照表

* [STUDENTS](#_5.学生表_STUDENTS)
* [TEACHERS](#_3.教师表_TEACHERS)
* [USERS](#_5-1.用户表_USERS)

### 6-6 “选择课程”用例

#### 1. 用例规约

| **用例名称** | **选择课程** |
| --- | --- |
| 功能 | 老师选择课程 |
| 参与者 | 老师 |
| 前置条件 | 必须先登录，选择课程必须是当前学期开课课程 |
| 后置条件 |  |
| 主事件流 | 1.查询当前学期开课课程  2.选择课程 |
| 备选事件流 | 1a.查询无课程    1.系统提示无课程可选择 2a. 课程已被选择     1.系统提示课程被选择     2. 老师重新选择。 |

| **用例名称** | **选择课程** |
| --- | --- |
| 功能 | 学生选择课程 |
| 参与者 | 学生 |
| 前置条件 | 必须先登录，选择课程必须是当前学期开课课程，课程已被老师选择 |
| 后置条件 |  |
| 主事件流 | 1.查询当前学期开课并已被教师选择的符合学生专业学期的课程  2.选择课程 |
| 备选事件流 | 1a.查询无课程    1.系统提示无课程可选择 2a. 课程已选择     1.系统提示课程已被选择，无需重选 |

#### 2. 业务流程

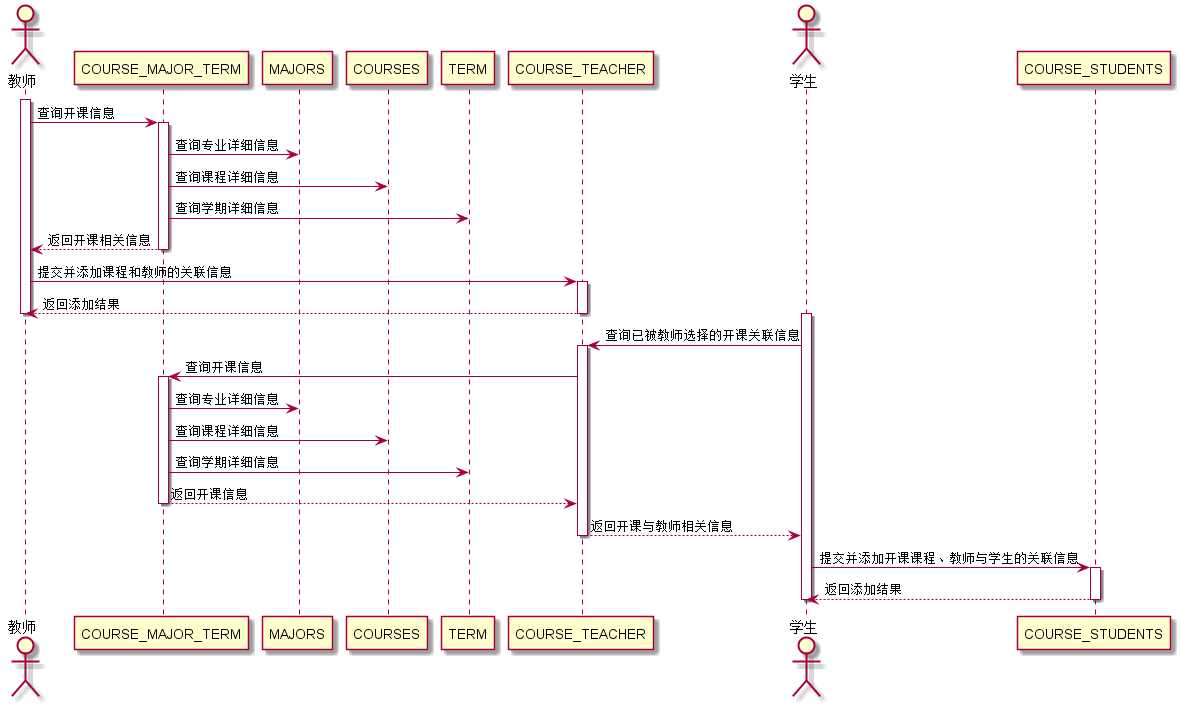


图 6-5

#### 3. 界面设计

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/选择课程.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E9%80%89%E6%8B%A9%E8%AF%BE%E7%A8%8B.html)
* API接口调用
  + 接口1：[getOpenCourseByTeacher](#_7-6_getOpenCourseByTeacher)
  + 接口2：[getOpenCourseByStudent](#_7-7_getOpenCourseByStudent)
  + 接口3：[setTeacherSelectCourse](#_7-8_setTeacherSelectCourse)
  + 接口4：[setStudentSelectCourse](#_7-9_setStudentSelectCourse)

#### 4. 算法描述

无

#### 5. 参照表

* [USERS](#_5-1.用户表_USERS)
* [TEACHERS](#_3.教师表_TEACHERS)
* [STUDENTS](#_5.学生表_STUDENTS)
* [MAJORS](#_6.专业表_MAJORS)
* [COURSES](#_7.课程表_COURSES)
* [TERMS](#_8.学期表_TERMS)
* [COURSE\_MAJOR\_TERM](#_10.课程专业学期关联表_COURSE_MAJOR_TERM)
* [COURSE\_TEACHER](#_9.开课课程教师关联表_COURSE_TEACHER)
* [COURSE\_STUDENTS](#_11.开课课程学生关联表_COURSE_STUDENTS)

### 6-7“查看课程”用例

#### 1. 用例规约

| **用例名称** | **查看课程** |
| --- | --- |
| 功能 | 查看用户当前学期已选择课程 |
| 参与者 | 学生/老师 |
| 前置条件 | 用户已登录 |
| 后置条件 |  |
| 主事件流 | 1.查询需要查看课程的相关信息 |
| 备选事件流 | 1a.查询数据出错    1.系统提示查询数据出错 |

#### 2. 业务流程

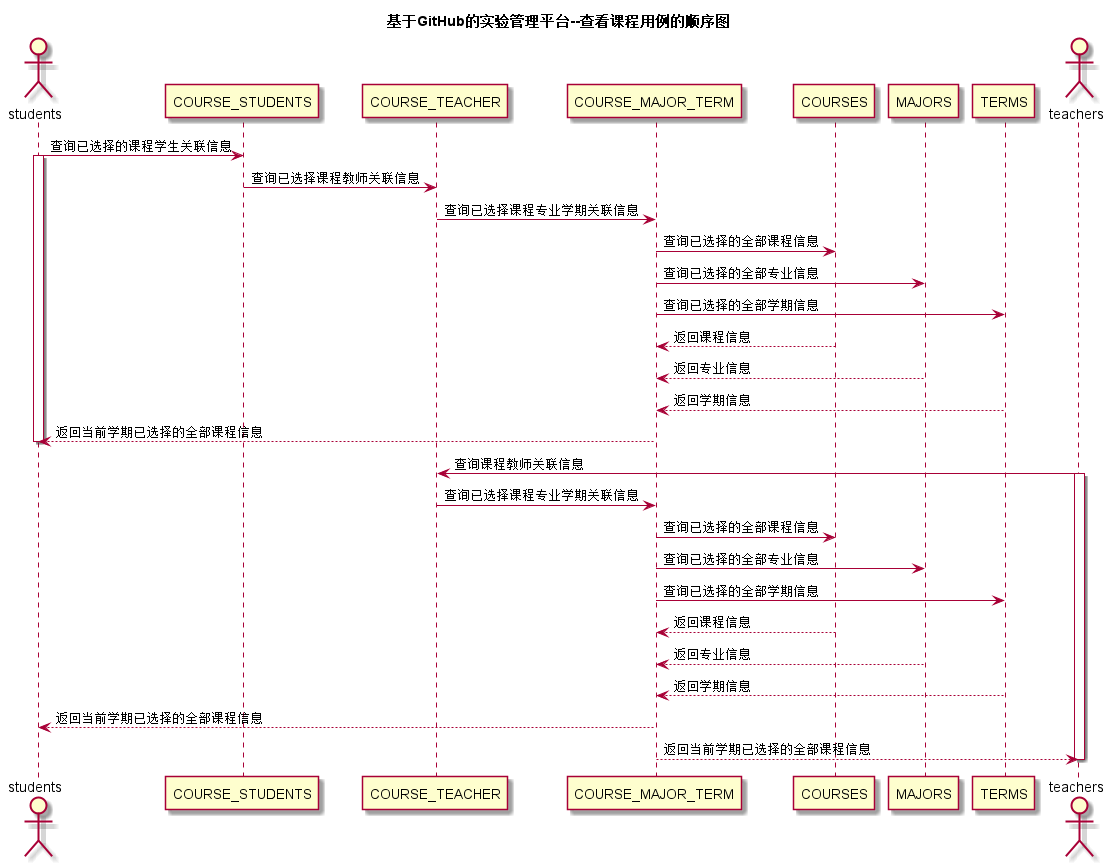


图 6-6

#### 3. 界面设计

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/查看课程.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E6%9F%A5%E7%9C%8B%E8%AF%BE%E7%A8%8B.html)
* API接口调用
  + 接口1：[getCourseByTeacher](#_7-10_getCourseByTeacher)
  + 接口2：[getCourseByStudent](#_7-11_getCourseByStudent)

#### 4. 算法描述

无

#### 5. 参照表

* [TEACHERS](#_3.教师表_TEACHERS)
* [STUDENTS](#_5.学生表_STUDENTS)
* [MAJORS](#_6.专业表_MAJORS)
* [COURSES](#_7.课程表_COURSES)
* [TERMS](#_8.学期表_TERMS)
* [COURSE\_MAJOR\_TERM](#_10.课程专业学期关联表_COURSE_MAJOR_TERM)
* [COURSE\_TEACHER](#_9.开课课程教师关联表_COURSE_TEACHER)
* [COURSE\_STUDENTS](#_11.开课课程学生关联表_COURSE_STUDENTS)

### 6-8 “查看学生信息”用例

#### 1. 用例规约

| **用例名称** | **查看学生信息** |
| --- | --- |
| 功能 | 查看当前实验下的学生信息 |
| 参与者 | 学生/老师 |
| 前置条件 |  |
| 后置条件 |  |
| 主事件流 | 1. 查询当前实验下的全部学生信息 |
| 备选事件流 | 1a.查询失败     1.系统提示查询失败 |

#### 2. 业务流程

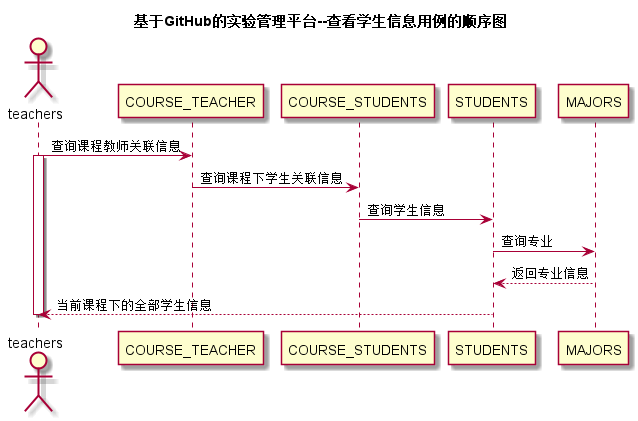


图 6-7

#### 3. 界面设计

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/查看学生信息.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E6%9F%A5%E7%9C%8B%E5%AD%A6%E7%94%9F%E4%BF%A1%E6%81%AF.html)
* API接口调用
  + 接口1：[getCourseTestAllStudents](#_7-12_getCourseTestAllStudents)

#### 4. 算法描述

无

#### 5. 参照表

* [STUDENTS](#_5.学生表_STUDENTS)
* [MAJORS](#_6.专业表_MAJORS)
* [COURSE\_MAJOR\_TERM](#_10.课程专业学期关联表_COURSE_MAJOR_TERM)
* [COURSE\_TEACHER](#_9.开课课程教师关联表_COURSE_TEACHER)
* [TESTS](#_12.实验表_TESTS)

### 6-9“查看实验”用例

#### 1. 用例规约

| **用例名称** | **查看实验** |
| --- | --- |
| 功能 | 查询当前课程下已发布的全部实验 |
| 参与者 | 学生/老师 |
| 前置条件 |  |
| 后置条件 |  |
| 主事件流 | 1. 查询实验信息 |
| 备选事件流 | 1a.查询失败     1.系统提示查询失败 |

#### 2. 业务流程

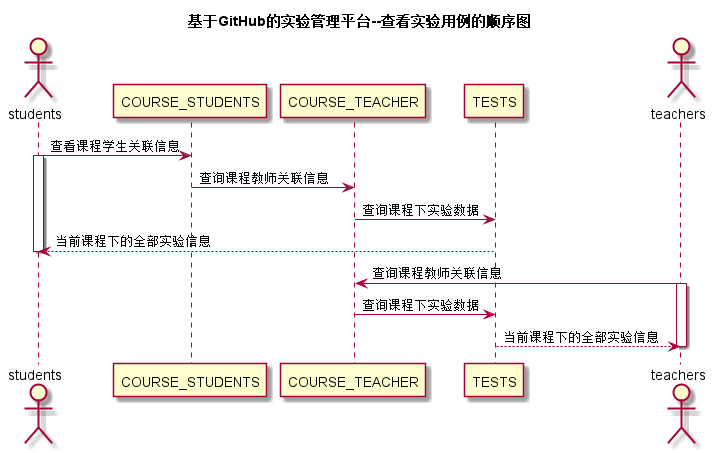


图 6-8

#### 3. 界面设计

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/查询实验.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E6%9F%A5%E8%AF%A2%E5%AE%9E%E9%AA%8C.html)
* API接口调用
  + 接口1：[getCourseAllTests](#_7-13_getCourseAllTests)

#### 4. 算法描述

无

#### 5. 参照表

* [COURSE\_TEACHER](#_9.开课课程教师关联表_COURSE_TEACHER)
* [COURSE\_STUDENTS](#_11.开课课程学生关联表_COURSE_STUDENTS)
* [TESTS](#_12.实验表_TESTS)

### 6-10“发布实验”用例

#### 1. 用例规约

| **用例名称** | **发布实验** |
| --- | --- |
| 功能 | 教师发布当前课程实验 |
| 参与者 | 老师 |
| 前置条件 | 必须先登录，并在查看实验页面下 |
| 后置条件 |  |
| 主事件流 | 1.教师填写课程实验及评价标准信息  2.教师提交实验信息  3.系统存储实验信息与评价标准信息 |
| 备选事件流 | 1a. 评价标准为空     1.系统提示添加评价标准信息 2a.提交失败     1.系统提示提交信息失败  2.重新提交 3a.系统存储失败     1.系统提示数据存储失败  2.重新提交 |

#### 2. 业务流程

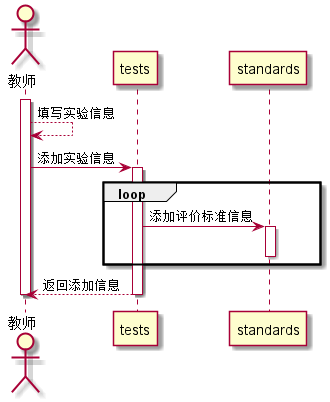


图 6-9

#### 3. 界面设计

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/修改用户信息.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E4%BF%AE%E6%94%B9%E7%94%A8%E6%88%B7%E4%BF%A1%E6%81%AF.html)
* API接口调用
  + 接口1：[setNewCourseTest](#_7-14_setNewCourseTest)

#### 4. 算法描述

无

#### 5. 参照表

* [TESTS](#_12.实验表_TESTS)
* [STANDARDS](#_13.评分标准表_STANDARDS)

### 6-11 “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）

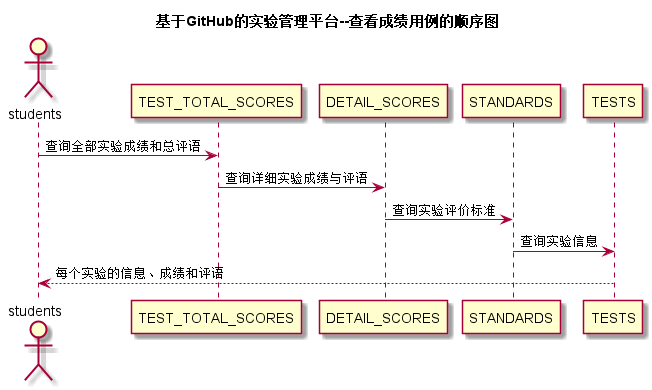


图 6-10

#### 3. 界面设计

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/查看成绩.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E6%9F%A5%E7%9C%8B%E6%88%90%E7%BB%A9.html)
* API接口调用
  + 接口1：[getOneStudentResults](#_7-15_getOneStudentResults)

#### 4. 算法描述

无

#### 5. 参照表

* [STUDENTS](#_5.学生表_STUDENTS)
* [TEST\_TOTAL\_SCORES](#_15.实验总评分表_TEST_TOTAL_SCORES)
* [DETAIL\_SCORES](#_14.详细评分表_DETAIL_SCORES)
* [STANDARDS](#_13.评分标准表_STANDARDS)
* [TESTS](#_12.实验表_TESTS)

### 6-12“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 | 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，系统自动设置成绩更新日期为当前日期，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详细评分项成绩和评语  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 |

#### 2. 业务流程（顺序图）

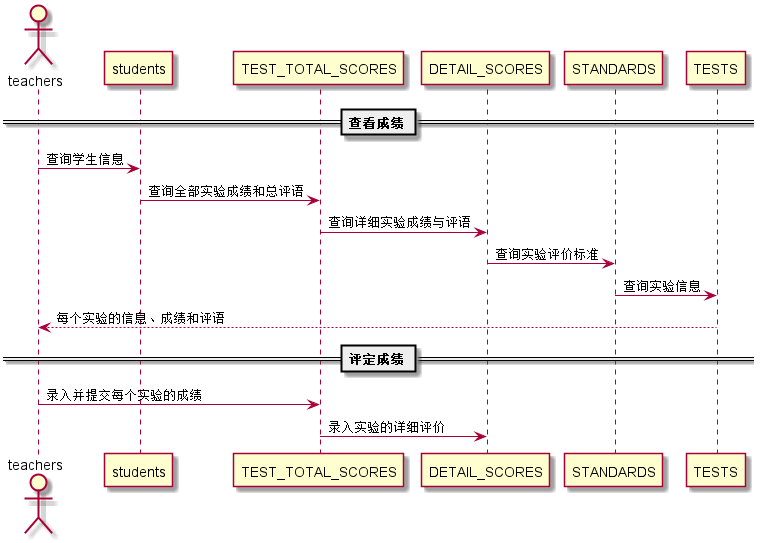


图 6-11

#### 3. 界面设计

* 界面参照: [https://Everythingtomyheart.github.io/is\_analysis\_pages/test6/ui/评定成绩.html](https://everythingtomyheart.github.io/is_analysis_pages/test6/ui/%E8%AF%84%E5%AE%9A%E6%88%90%E7%BB%A9.html)
* API接口调用
  + 接口1：[getOneStudentResults](#_7-15_getOneStudentResults)

用于显示一个学生的所有实验成绩和评语

* + 接口2：[setOneStudentResults](#_7-16_setOneStudentResults)

用于设置一个学生的部分实验成绩和评语

#### 4. 算法描述

无

#### 5. 参照表

* [STUDENTS](#_5.学生表_STUDENTS)
* [TEST\_TOTAL\_SCORES](#_15.实验总评分表_TEST_TOTAL_SCORES)
* [DETAIL\_SCORES](#_14.详细评分表_DETAIL_SCORES)
* [STANDARDS](#_13.评分标准表_STANDARDS)
* [TESTS](#_12.实验表_TESTS)

## 7.接口描述

### 7-1 login

用例： [登录](登录#_6-1)

* 功能： 登录到平台。
* 权限： 访客。
* API请求地址： 接口基本地址/v1/api/login
* 请求方式 ： POST
* 请求实例：

{

"id":"21048329823",

"password":"ABCDE",

"type":"1"//1,2, 1为学生，2为教师

}

* 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，不是原文，是加密后的字符串 |
| type | 用户类型，学生或者老师 |

* 返回实例：

{

"code":200,

"msg":"登录成功",

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |

### 7-2 loginout

用例： [登出](登出#_6-2_)

* 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/loginout
* 请求方式 ： POST
* 请求实例：

{

"user\_id":324111

}

* 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USERS.USER\_ID的值 |

* 返回实例：

{

"code":200,

"msg": "注销成功",

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |

### 7-3 setPassword

用例： [修改密码](修改密码#_6-3)

* 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,

"oldPassword":awrsdtfghjk765432,

"password":asasdfgh,

}

* 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 |
| oldPassword | 用户的旧密码。不能为空，必须经过加密码,不能是密码的原文。 |
| password | 用户的密码。不能为空，必须经过加密码,不能是密码的原文。 |

* 返回实例：

{

"code": 200,

"msg":修改成功

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |

### 7-4 updateUserInfo

用例： [修改用户信息](修改用户信息#_6-4_)

* 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/updateUserInfo
* 请求方式 ： POST
* 请求实例：

{

"user\_id":"21048329823",

"github\_username":"xxxx",

}

* 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。 |

* 返回实例：

{

"code": 200,

"msg": "修改成功"

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |

### 7-5 getUserInfo

用例： [查看用户信息](查看用户信息#_6-5),[修改用户信息](修改用户信息#_6-4_)

* 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:

无

* 返回实例：

{

"code": 200,

"msg": "查询成功",

data:{

"ID":"2101201281",

"name":"张三",

"grade\_class":"2016级4班"

"github\_username":"ABCD",

"type":"学生"，

"course":"软件工程"

}

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |
| data | 查询到的相关信息 |

### 7-6 getOpenCourseByTeacher

用例： [选择课程](选择课程#_6-6_)

* 功能： 查看当前学期开课信息
* 权限： 教师
* API请求地址： 接口基本地址/v1/api/getOpenCourseByTeacher
* 请求方式 ： POST
* 请求实例： 无
* 请求参数说明: 无
* 返回实例：

{

"code":200,

"msg":"选择成功",

"data":[{

course\_major\_term\_id:"123",

"course\_name":"软件",

}]

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |
| data | 查询到的相应信息 |

### 7-7 getOpenCourseByStudent

用例： [选择课程](选择课程#_6-6_)

* 功能： 查看当前学期开课并已被教师选择课程
* 权限： 学生
* API请求地址： 接口基本地址/v1/api/getOpenCourseByStudent
* 请求方式 ： POST
* 请求实例： 无
* 请求参数说明:
* 返回实例：

{

"code":200,

"msg":"选择成功",

"data":[{

course\_teacher\_id:"123",

"course\_name":"软件",

"teacher\_name":"xxx"

}]

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |
| data | 查询到的相应信息 |

### 7-8 setTeacherSelectCourse

用例： [选择课程](选择课程#_6-6_)

* 功能： 选择教师选中课程
* 权限： 教师
* API请求地址： 接口基本地址/v1/api/setTeacherSelectCourse
* 请求方式 ： POST
* 请求实例：

{

"course\_major\_term\_id":123

}

* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| course\_major\_term\_id | 当前学期开课课程id |

* 返回实例：

{

"code":200,

"msg":"选择成功",

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |

### 7-9 setStudentSelectCourse

用例： [选择课程](选择课程#_6-6_)

* 功能： 选择学生选中开课课程
* 权限： 学生
* API请求地址： 接口基本地址/v1/api/setStudentSelectCourse
* 请求方式 ： POST
* 请求实例：

{

"course\_teacher\_id":123

}

* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| course\_teacher\_id | 选中的教师开课信息id |

* 返回实例：

{

"code":200,

"msg":"选择成功",

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |

### 7-10 getCourseByTeacher

用例： [查看课程](查看课程#_6-7)

* 功能： 查看教师自身已选择课程
* 权限： 教师
* API请求地址： 接口基本地址/v1/api/getCourseByTeacher
* 请求方式 ： POST
* 请求实例： 无
* 请求参数说明: 无
* 返回实例：

{

"code":200,

"msg":"选择成功",

"data":[{

course\_teacher\_id:"123",

"course\_name":"软件",

"major":"软件工程",

"grade\_class":"2016级3、4班"

}]

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |
| data | 查询到的相应信息 |

### 7-11 getCourseByStudent

用例： [查看课程](查看课程#_6-7)

* 功能： 查看学生自身已选择课程
* 权限： 学生
* API请求地址： 接口基本地址/v1/api/getCourseByStudent
* 请求方式 ： POST
* 请求实例： 无
* 请求参数说明: 无
* 返回实例：

{

"code":200,

"msg":"选择成功",

"data":[{

course\_student\_id:"123",

"course\_name":"软件",

"major":"软件工程",

"teacher\_name":"XXX"

}]

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |
| data | 查询到的相应信息 |

### 7-12 getCourseTestAllStudents

用例： [查看学生信息](查看学生信息#_6-8_)

* 功能： 查看当前课程实验下的全部学生信息
* 权限： 教师、学生
* API请求地址： 接口基本地址/v1/api/getCourseTestAllStudents
* 请求方式 ： POST
* 请求实例： 无
* 请求参数说明:
* 返回实例：

{

"code":200,

"msg":"查询成功",

"data":[{

"student\_id":"123",

"student\_name":"肖清宇",

"grade\_class":"2016级4班"

}]

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |
| data | 查询到的相应信息 |

### 7-13 getCourseAllTests

用例： [查看实验](查看实验#_6-9)

* 功能： 查看当前课程全部实验
* 权限： 学生，教师
* API请求地址： 接口基本地址/v1/api/getCourseByStudent/<course\_teacher\_id>
* 请求方式 ： get
* 请求实例： { course\_teacher\_id:"123" }
* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| course\_teacher\_id | 开课课程与教师关联信息id |

* 返回实例：

{

"code":200,

"msg":"选择成功",

"data":[{

test\_id:"123",

"test\_name":"实验1",

}]

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |
| data | 查询到的相应信息 |

### 7-14 setNewCourseTest

用例： [发布实验](发布实验#_6-10)

* 功能： 教师发布新实验
* 权限： 老师，需先登录
* API请求地址： 接口基本地址/v1/api/setNewCourseTest
* 请求方式 ： POST
* 请求实例：

{

"test\_name":"xxxx",

"test\_url":"xxxx",

"course\_teacher\_id":"xxxx",

"standards":[{

"standard\_name":"xxx",

"standard\_detail":"xxx",

"standard\_mix":20

}]

}

* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| test\_name | 实验名称 |
| test\_url | 实验描述的github连接 |
| course\_teacher\_id | 关联开课课程、教师关联信息id |
| standards | 评价标准 |
| standard\_name | 评价标准名称 |
| standard\_detail | 评价标准描述 |
| standard\_mix | 占总成绩比例 |

* 返回实例：

{

"code": 200,

"msg": "添加成功"

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |

### 7-15 getOneStudentResults

用例： [查看成绩](查看成绩#_6-11_)，[评定成绩](评定成绩#_6-12)

* 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>&&<course\_teacher\_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 |
| course\_teacher\_id | 关联课程编号 |

* 返回实例：

{

"code":200,

"msg":"查询成功",

"data":{

"student\_id": "201510315203",

"github\_username": "chinajuedui",

"major\_grade\_class": "软件(本)16-4",

"name": "肖清宇",

"total": 6,

"avgresult":90.5,

"test\_data": [

{

"total\_score\_id":1,

"total\_score": 91,

"test\_name":"实验1",

"test\_url":"xxxx",

"total\_memo":"本实验做得好",

"update\_date": "2018-04-02 13:48:01",

web\_exists:"true",

"detail\_scores":[

"score":60,

"detail\_memo":"文档存在一定问题",

"standard\_name":"文档规范",

"standard\_detail":"文档必须使用md格式编写，格式规范",

"standard\_mix":"40"

},

...其他评分项

]

},

{

...其他实验

}

]

}

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |
| data | 查询到的相应信息 |
| student\_id | 学号 |
| github\_username | 学生的gitHub用户名 |
| major\_grade\_class | 专业班级 |
| name | 真实姓名 |
| total | 实验总数 |
| avgresult | 实验平均成绩 |
| test\_data | 所有实验的成绩和评语 |
| total\_score\_id | 总评分编号 |
| test\_name | 实验名称 |
| test\_url | 实验要求url |
| web\_exists | 本实验的GitHub网页是否存在 |
| total\_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。 |
| total\_memo | 本实验老师的总评价，可以为空 |
| update\_date | 本实验老师的批改日期，可以为空 |
| detail\_scores | 详细评分项 |
| score | 详细评分 |
| detail\_memo | 详细评分项评价 |
| standard\_name | 评分标准名称 |
| standard\_detail | 评分标准描述 |
| standard\_mix | 评分占比 |

### 7-16 setOneStudentResults

用例： [评定成绩](评定成绩#_6-12)

* 功能： 设置一个学生的部分实验成绩和评语。

输入参数result不为空，自动设置update\_date为当前日期，表示同时设置批改日期。

输入参数result为空，自动设置update\_date为空，表示未批改。

* 权限： 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/etOneStudentResults
* 请求方式 ： POST

- 请求实例：

{

"student\_id": "201510315203",

"total": 6,

"test\_data": [

{

"total\_score\_id":1,

"total\_score": 91,

"total\_memo":"本实验做得好",

"detail\_scores":[

"score":60,

"detail\_memo":"文档存在一定问题",

"standard\_mix":"40"

},

...其他评分项

]

},

{

...其他实验

}

]

}

* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| student\_id | 学号 |
| github\_username | 学生的gitHub用户名 |
| major\_grade\_class | 专业班级 |
| name | 真实姓名 |
| total | 实验总数 |
| test\_data | 所有实验的成绩和评语 |
| total\_score\_id | 总评分编号 |
| total\_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。 |
| total\_memo | 本实验老师的总评价，可以为空 |
| update\_date | 本实验老师的批改日期，可以为空 |
| detail\_scores | 详细评分项 |
| score | 详细评分 |
| detail\_memo | 详细评分项评价 |
| standard\_mix | 评分占比 |

* 返回实例：

{

"code": 200,

"msg": "评定成功"

}

* 返回参数说明：

| **参数名称** | **说明** |
| --- | --- |
| code | 返回结果状态码 |
| msg | 返回结果说明信息 |