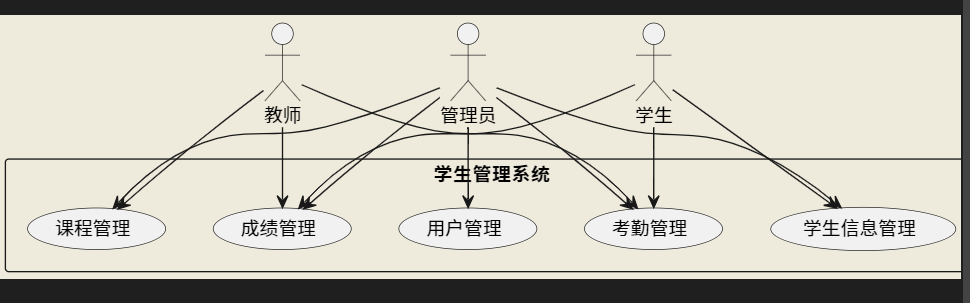
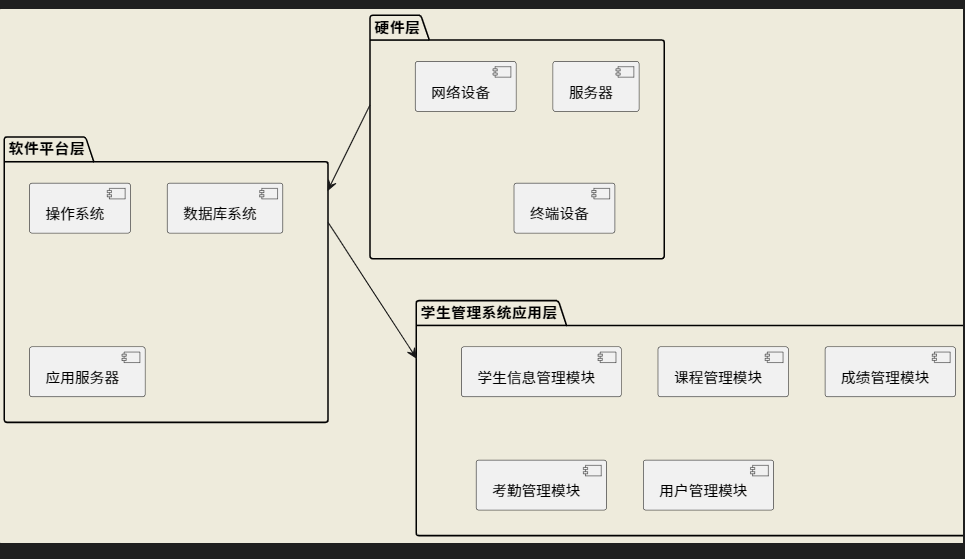
**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于jQuery的学生管理系统的分析与设计 | | |
| 课 程 | 软件工程 | | |
| 学 院 | 信息工程学院 | | |
| 专 业 | 软件技术 | 年级 | 2024级 |
| 学生姓名 | 赵梓锐 | 学号 | 112322100004 |

|  |  |  |  |
| --- | --- | --- | --- |
| **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，内容: 可行性分析、需求分析、概要设计、详细设计。 | 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** | | |  |

2024年12月22日

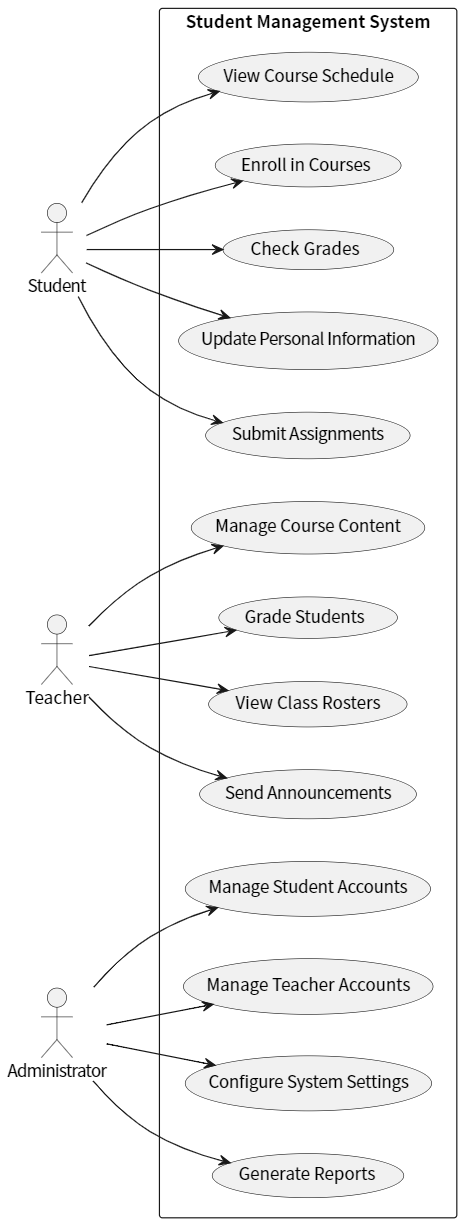
# 2. 系统总体结构



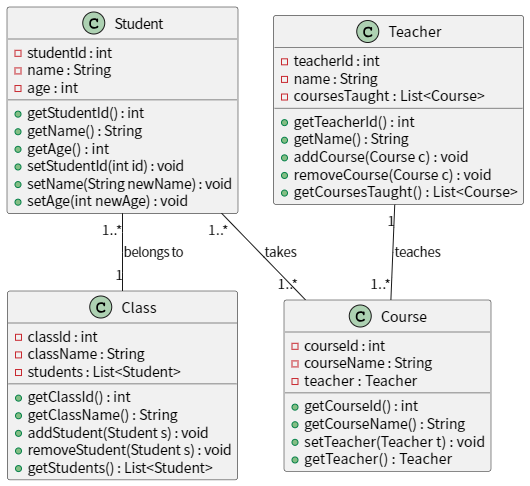
登录用例



# 3. 用例图设计

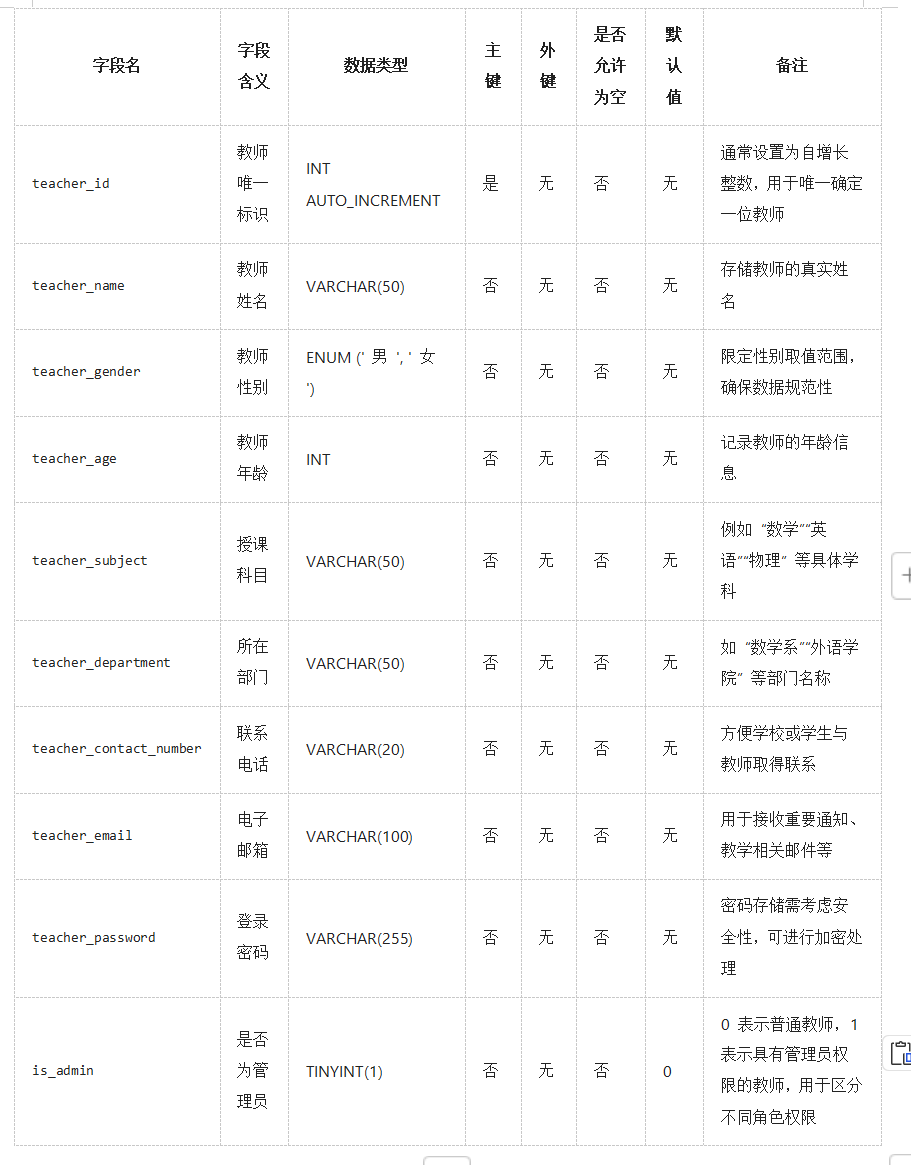


4. 类图设计

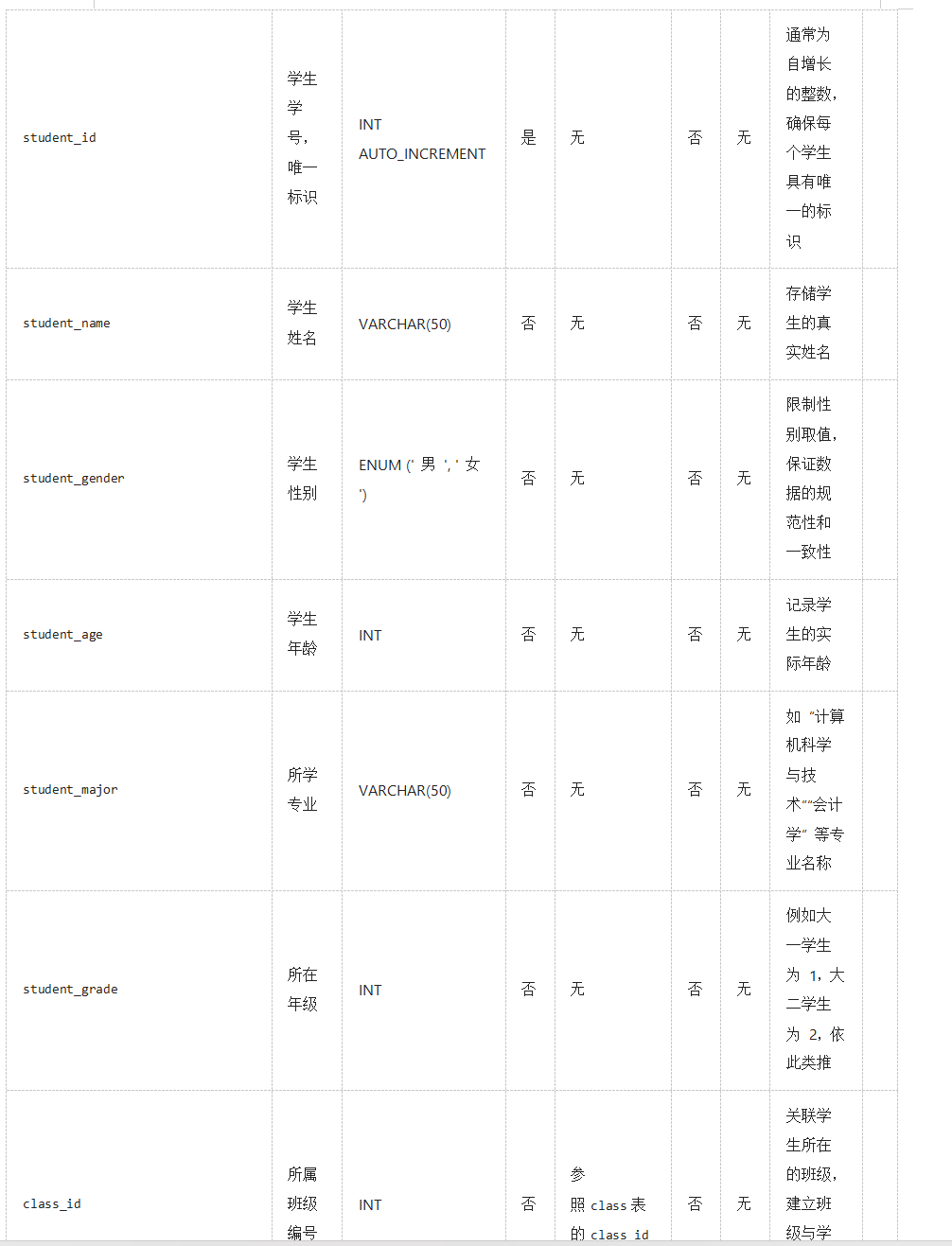


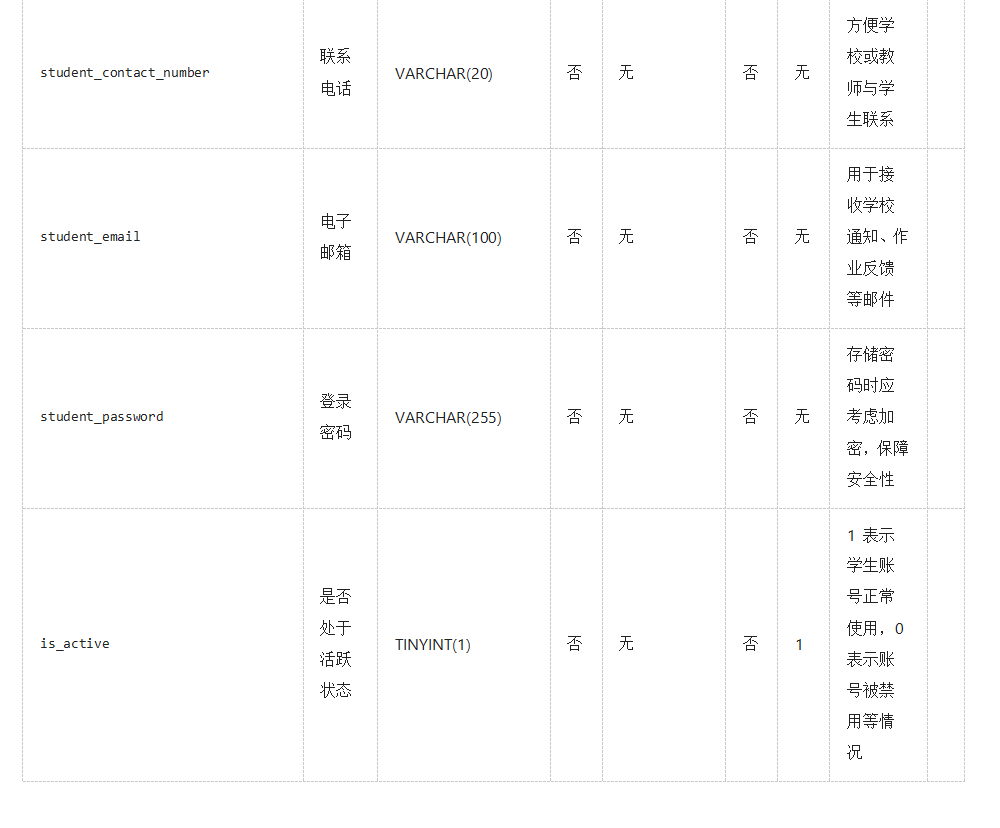
# 5. 数据库设计

## TEACHERS表（老师表）



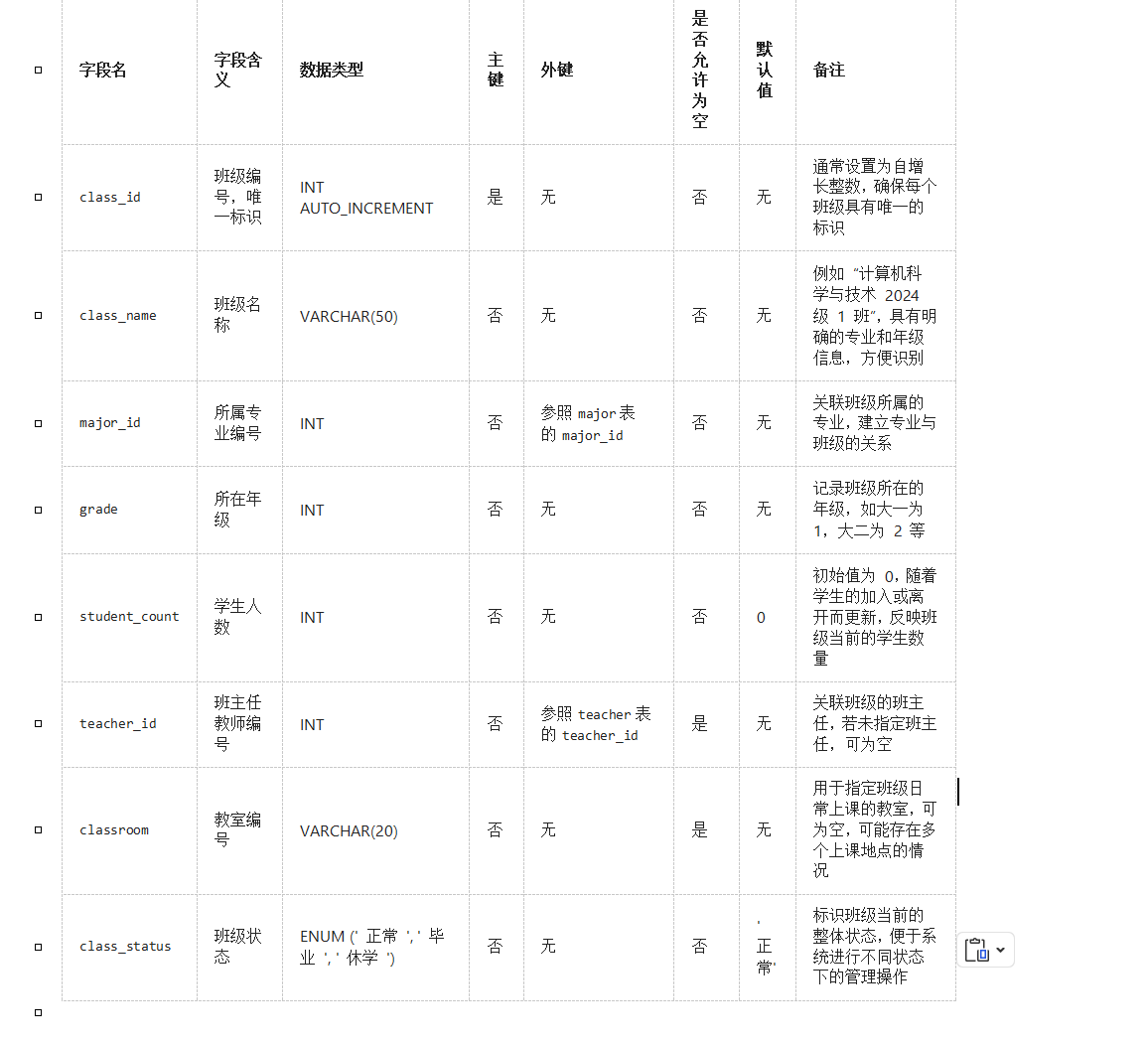
STUDENTS表（学生表）



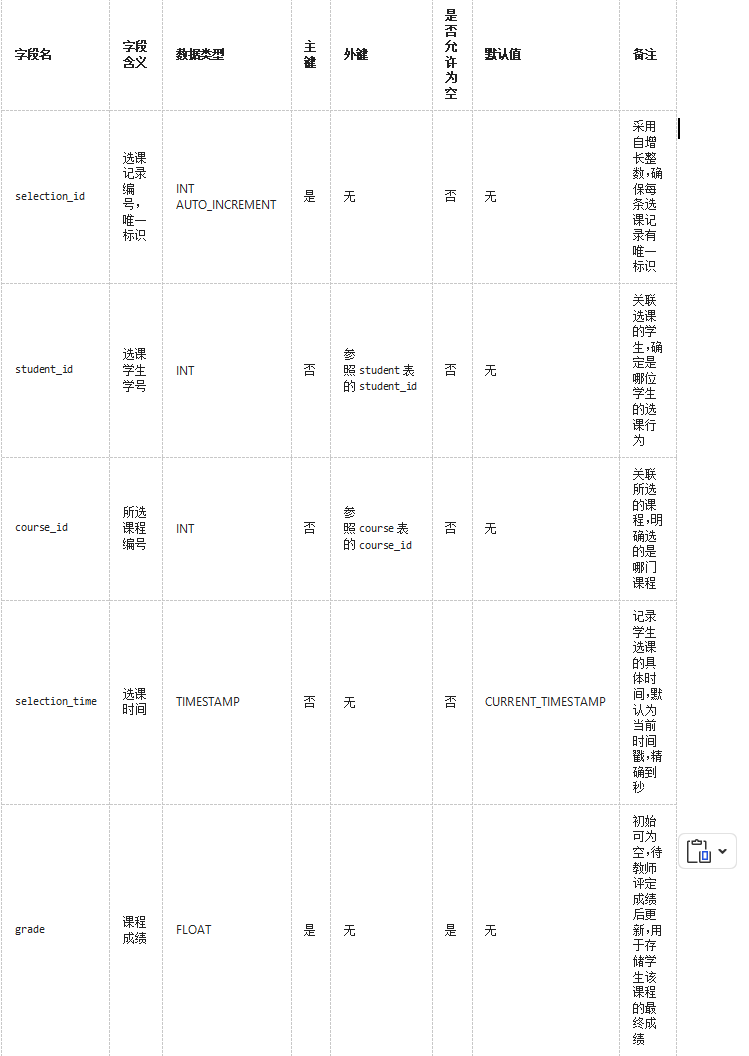


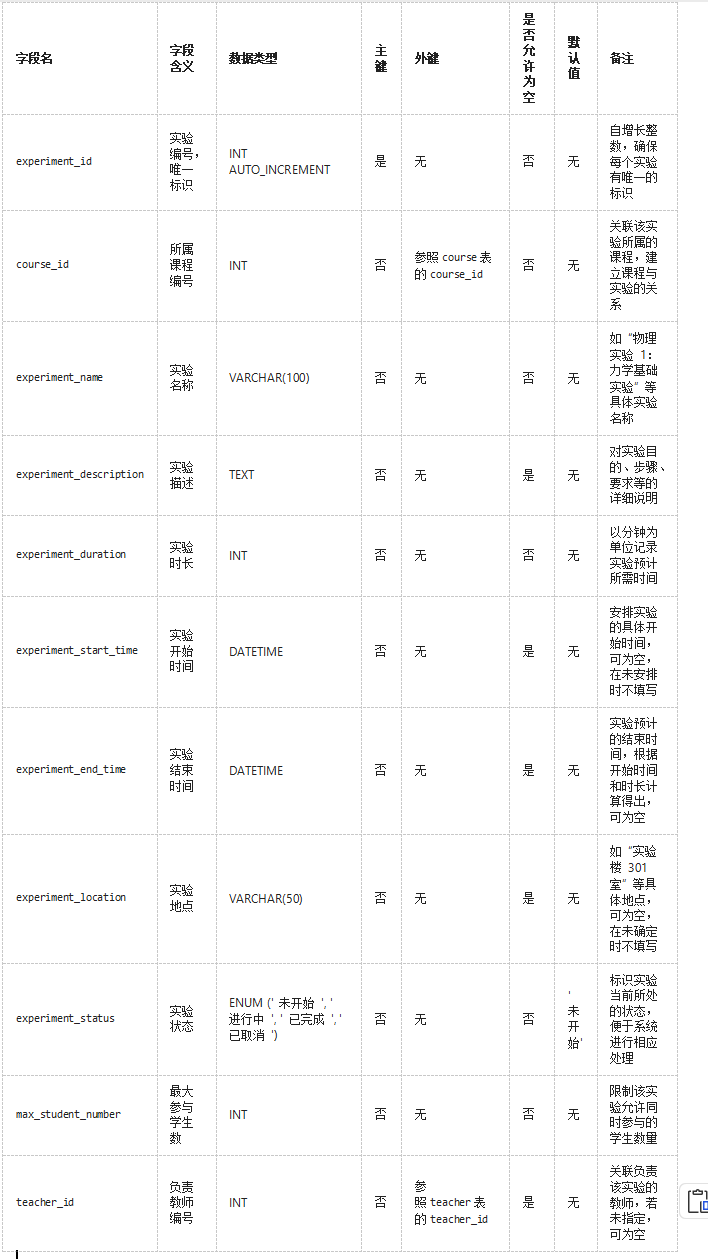
## LESSON表（课程表）屏幕截图 2024-12-22 154746屏幕截图 2024-12-22 154741

CLASS表（班级表）



## CHOOSE\_LESSON表（选课表）



TESTS表（实验项目表）

## GRADES表（学生实验成绩表）

| **字段名** | **字段含义** | **数据类型** | **主键** | **外键** | **是否允许为空** | **默认值** | **备注** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| grade\_id | 成绩记录编号，唯一标识 | INT AUTO\_INCREMENT | 是 | 无 | 否 | 无 | 自增长整数，确保每条成绩记录有唯一标识 |
| student\_id | 学生学号 | INT | 否 | 参照 student 表的 student\_id | 否 | 无 | 关联对应的学生，确定是哪位学生的成绩 |
| experiment\_id | 实验编号 | INT | 否 | 参照 experiment 表的 experiment\_id | 否 | 无 | 关联对应的实验，明确是哪次实验的成绩 |
| grade | 实验成绩 | FLOAT | 否 | 无 | 否 | 无 | 记录学生该实验的成绩，可根据实际评分标准设定范围，如 0 - 100 分 |
| grade\_comment | 成绩评语 | VARCHAR(200) | 否 | 无 | 是 | 无 | 教师对学生实验表现的评语，可为空 |
| submission\_time | 成绩提交时间 | TIMESTAMP | 否 | 无 | 否 | CURRENT\_TIMESTAMP | 记录教师提交成绩的时间，默认为当前时间戳 |

## AVERAGE\_SCORE表（实验平均成绩表）

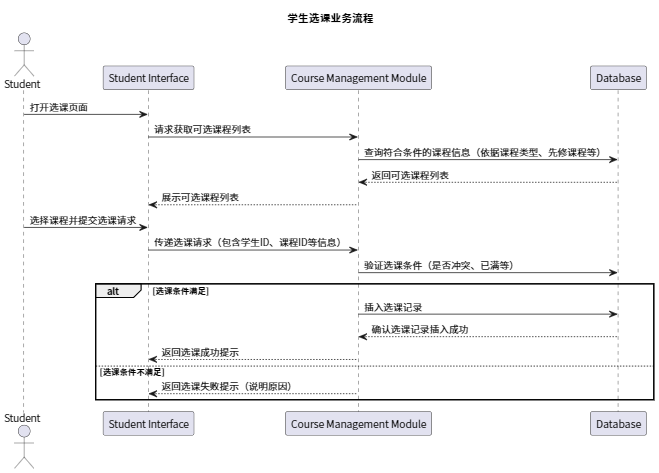
| **字段名** | **字段含义** | **数据类型** | **主键** | **外键** | **是否允许为空** | **默认值** | **备注** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| average\_grade\_id | 平均成绩记录编号，唯一标识 | INT AUTO\_INCREMENT | 是 | 无 | 否 | 无 | 自增长整数，用于区分每条平均成绩记录 |
| experiment\_id | 实验编号 | INT | 否 | 参照 experiment 表的 experiment\_id | 否 | 无 | 关联对应的实验，确定是哪次实验的平均成绩 |
| average\_grade | 实验平均成绩 | FLOAT | 否 | 无 | 否 | 无 | 存储该实验所有学生成绩的平均值，保留一定小数位数，如两位小数 |
| total\_student\_count | 参与学生总数 | INT | 否 | 无 | 否 | 无 | 记录参与该实验的学生数量，用于计算平均成绩和统计分析 |
| grade\_statistics | 成绩统计信息（如各分数段人数分布） | VARCHAR(200) | 否 | 无 | 是 | 无 | 以特定格式存储成绩统计情况，例如 “0 - 59 分：5 人；60 - 79 分：10 人；80 - 100 分：8 人”，方便后续分析成绩分布情况 |
| update\_time | 数据更新时间 | TIMESTAMP | 否 | 无 | 否 | CURRENT\_TIMESTAMP | 记录该平均成绩数据的最后更新时间，默认为当前时间戳，每次成绩有变化时自动更新 |

6. 用例及界面详细设计

6.1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。 |

6.2业务流程（顺序图）

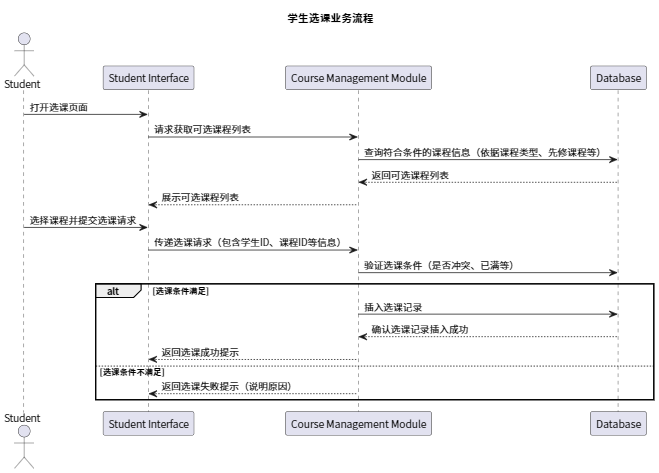


### 6.3界面设计



### 6.4接口设计

### 6.4.2业务流程（顺序图）



### 6.4.4接口设计

教师查看学生成绩接口

### **接口路径**

/api/v1/teacher/student-grades

### **请求方式**

GET

### **请求头**

Authorization（字符串，必选）：教师登录成功后获取的身份验证令牌（Token），用于验证教师的身份合法性，确保只有已登录且具备相应权限的教师能够访问该接口。

### **请求参数**

course\_id（整数，可选）：要查看成绩的具体课程编号。如果教师不传该参数，则默认返回该教师所授所有课程下学生的成绩信息；若传入该参数，则仅返回指定课程对应的学生成绩情况。

student\_id（整数，可选）：学生的学号，用于进一步筛选查看特定学生在某门课程（如果传入 course\_id）或者所有课程（如果未传入 course\_id）中的成绩信息。若不传该参数，则返回符合课程筛选条件（若有）下的所有学生成绩。

### **响应结果**

****成功获取（状态码 200）****：

message（字符串）：“成绩查询成功”。

data（对象）：包含成绩相关的数据结构，具体如下：

course\_info（对象，可选）：如果请求时传入了 course\_id 参数，这里会展示该课程的详细信息，例如 course\_name（课程名称）、course\_credits（课程学分）、teacher\_id（授课教师编号，即当前登录教师的编号）等。

student\_grades（数组，元素为对象）：包含每位学生的成绩详细信息，每个对象的结构如下：

student\_id（整数）：学生的学号，用于唯一标识学生。

student\_name（字符串）：学生的姓名，方便教师直观了解成绩对应的学生是谁。

grade（浮点数）：学生在对应课程（若有 course\_id 参数指定）或所有课程（若未指定 course\_id）中的成绩分数，成绩的取值范围和评分标准可根据学校实际规定设定，例如 0 - 100 分。

assessment\_details（字符串，可选）：成绩评定的详细说明，比如平时成绩占比、考试成绩占比以及各项成绩的具体分值等，帮助教师和学生清楚了解成绩的构成情况。

****身份验证失败（状态码 401）****：

message（字符串）：“身份验证失败，请重新登录”。

****参数错误（状态码 400）****：

message（字符串）：“请求参数错误，例如 course\_id 或 student\_id 格式不正确、不存在等，请检查后重新请求”，具体说明参数不符合要求的原因。

### **功能描述**

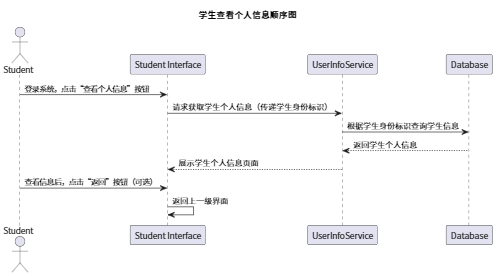
该接口旨在为教师提供便捷的途径来查看学生成绩信息。教师可以根据自己的需求，通过传入不同的 course\_id 和 student\_id 参数，灵活地查看指定课程下所有学生、特定学生在所有课程或者特定学生在特定课程中的成绩情况，方便教师进行教学效果评估、成绩分析以及后续的教学调整等工作，同时系统会对教师的身份进行严格验证，保障数据的安全性和隐私性，仅允许合法登录的教师访问相应的成绩数据。

## 6.5 [“查看用户信息”用例](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E7%94%A8%E4%BE%8B\\%E6%9F%A5%E7%9C%8B%E7%94%A8%E6%88%B7%E4%BF%A1%E6%81%AF.md),[界面](https://ssw383318348.github.io/is_analysis_pages/test6/%E6%9F%A5%E7%9C%8B%E7%94%A8%E6%88%B7%E4%BF%A1%E6%81%AF.html)

6.5.1用例规约

| **用例编号** | **UC01** |
| --- | --- |
| 用例名称 | 查看用户信息 |
| 执行者 | 学生、教师、管理员 |
| 前置条件 | 用户已成功登录到学生管理系统 |
| 基本事件流 | ****学生查看个人信息****： 1. 学生登录后，在系统主界面或个人中心找到 “查看个人信息” 按钮并点击。 2. 系统从数据库中查询该学生的详细信息，包括学号、姓名、性别、年龄、专业、年级、班级、联系方式等。 3. 系统在新页面以清晰的表格形式展示学生的个人信息，各字段标注明确。 4. 页面提供 “返回” 按钮，学生点击后可返回上一级界面。 ****教师查看个人信息和所授班级学生信息****： 1. 教师登录后，点击 “查看个人信息” 按钮，进入个人信息查看页面，该页面展示教师的编号、姓名、性别、年龄、所授科目、所在部门、联系方式等信息。 2. 在教师个人信息查看页面，提供 “查看所授班级学生信息” 链接，教师点击后，系统查询并展示所授班级学生的信息列表，包括学生的学号、姓名、性别、年龄、专业、年级等，列表支持搜索和筛选功能（如按学号、姓名、专业筛选）。 3. 教师个人信息查看页面和学生信息列表页面均提供 “返回” 按钮，用于返回上一级界面。 ****管理员查看和管理所有用户信息****： 1. 管理员登录后，进入 “用户信息管理” 模块，系统默认展示所有学生和教师的简要信息列表（学生显示学号、姓名、专业、年级；教师显示教师编号、姓名、所授科目、所在部门），列表支持搜索和筛选功能（如按用户类型、专业、部门筛选）。 2. 管理员点击某一用户的行或 “查看详情” 按钮，系统弹出该用户的详细信息弹窗或跳转到详细信息页面，展示该用户的所有信息，包括基本信息（与学生、教师查看自己信息时的字段相同）以及学生的选课记录、成绩信息、教师的授课安排等扩展信息（根据用户类型展示相应扩展信息），并在页面中提供 “编辑”“删除” 等操作按钮（对于学生用户，还有 “添加成绩”“查看考勤” 等操作按钮；对于教师用户，有 “安排课程” 等操作按钮）。 |

6.5.2业务流程（顺序图）



### 6.5.3界面设计



### 6.5.4接口设计

接口1：[getUserInfo](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E6%8E%A5%E5%8F%A3\\setPassword.md)

用例： [查看用户信息](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E7%94%A8%E4%BE%8B\\%E4%BF%AE%E6%94%B9%E5%AF%86%E7%A0%81.md)

* 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,

"ID":"2101201281",

"name":"张三",

"class\_dept":"软件工程1班"

"github\_username":"ABCD",

"type":"学生"

}

* 返回参数说明：

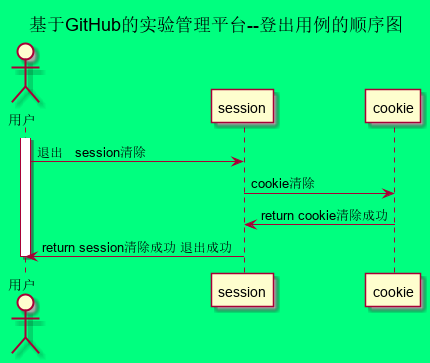
| **参数名称** | **说明** |
| --- | --- |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |
| ID | 学号或者工号 |
| name | 用户的真实姓名 |
| class\_dept | 班级或者部门名称 |
| github\_username | gitHub用户名 |
| type | 用户类型：老师或者学生 |

## 6.6 [“登出”用例](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E7%94%A8%E4%BE%8B\\%E7%99%BB%E5%87%BA.md),[界面](https://ssw383318348.github.io/is_analysis_pages/test6/%E7%99%BB%E5%87%BA.html)

### 6.6.1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 |

### 6.6.2业务流程（顺序图）



### 6.6.3界面设计



### 6.6.4接口设计

接口1：[logout](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E6%8E%A5%E5%8F%A3\\setPassword.md)

用例： [登出](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E7%94%A8%E4%BE%8B\\%E4%BF%AE%E6%94%B9%E5%AF%86%E7%A0%81.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":1234,

}

* 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

}

* 返回参数说明：

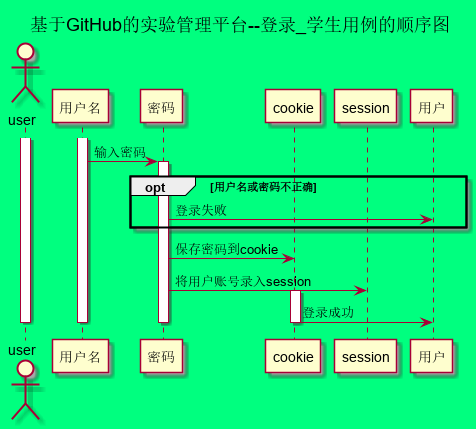
| **参数名称** | **说明** |
| --- | --- |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |

## 6.7 [“登录\_学生”用例](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E7%94%A8%E4%BE%8B\\%E7%99%BB%E5%BD%95.md),[界面](https://ssw383318348.github.io/is_analysis_pages/test6/%E7%99%BB%E5%BD%95_%E5%AD%A6%E7%94%9F.html)

### 6.7.1用例规约

| **用例名称** | **登录\_学生** |
| --- | --- |
| 功能 | 登录平台 |
| 参与者 | 学生 |
| 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 |

### 6.7.2业务流程（顺序图）



### 6.7.3界面设计



### 6.7.4接口设计

接口1：[login](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E6%8E%A5%E5%8F%A3\\setPassword.md)

用例： [登录](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E7%94%A8%E4%BE%8B\\%E4%BF%AE%E6%94%B9%E5%AF%86%E7%A0%81.md)

* 功能： 登录到平台。
* 权限： 学生/老师
* 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":"学生/老师"

}

* 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：

{

"status": true,

"info": null,

}

* 返回参数说明：

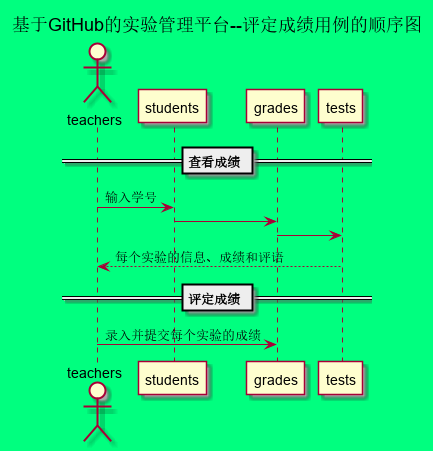
| **参数名称** | **说明** |
| --- | --- |
| status | bool类型，true表示正确的返回，false表示有错误 |
| info | 返回结果说明信息 |

## 6.9 [“评定成绩”用例](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E7%94%A8%E4%BE%8B\\%E8%AF%84%E5%AE%9A%E6%88%90%E7%BB%A9.md)

### 6.9.1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 |

### 6.9.2业务流程（顺序图）



### 6.9.3界面设计



### 6.9.4接口设计

接口1：[getNextPrevStudent](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E6%8E%A5%E5%8F%A3\\setPassword.md)

用例： [评定成绩](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E7%94%A8%E4%BE%8B\\%E4%BF%AE%E6%94%B9%E5%AF%86%E7%A0%81.md)

* 功能： 返回一个学生的上一个或者下一个学生的学号。
* 权限：只有老师可以调用该API。
* API请求地址：接口基本地址/v1/api/getNextPrevStudent/<is\_next>/<student\_id>
* 请求方式 ： GET
* 请求参数说明:

| **参数名称** | **说明** |
| --- | --- |
| is\_next | 是下一个还是上一个，如果为1表示取下一个学生，如果为0表示上一个学生 |
| student\_id | 学生的学号 |

* 返回实例：

{

"status": true,

"info": null,

"student\_id": "201510315203"

}

* 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 |

接口2： getOneStudentResults

用例： [查看成绩、评定成绩](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E7%94%A8%E4%BE%8B\\%E4%BF%AE%E6%94%B9%E5%AF%86%E7%A0%81.md)

* 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/getNextPrevStudent/ <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：

{

"status": true,

"info": null,

"student\_id": "201510315203",

"github\_username": "chinajuedui",

"class": "软件(本)15-1",

"name": "陈松华",

"total": 6,

"avgresult":90.5,

"data": [

{

"test\_id":1,

"web\_exists": true,

"result": 91,

"memo":"本实验做得好",

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

},

{

...其他实验

}

]

}

* 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 |
| avgresult | 实验平均成绩 |
| data | 所有实验的成绩和评语 |
| test\_id | 实验编号 |
| web\_exists | 本实验的GitHub网页是否存在 |
| result | 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。 |
| memo | 本实验老师的评价，可以为空 |
| update\_date | 本实验老师的批改日期，可以为空 |

接口3：[setOneStudentResults](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E6%8E%A5%E5%8F%A3\\setPassword.md)

用例： [评定成绩](file:///D:\\CDUcourse_isAnalysis\\100ssw\\test6\\%E7%94%A8%E4%BE%8B\\%E4%BF%AE%E6%94%B9%E5%AF%86%E7%A0%81.md)

* 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：只有老师可以调用该API。
* API请求地址：接口基本地址/v1/api/ setOneStudentResults
* 请求方式 ： POST
* 请求实例：

{

"student\_id": "201510315203",

"total": 6,

"data": [

{

"test\_id":1,

"result": 91,

"memo":"本实验做得好",

},

{

...其他实验

}

]

}

* 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 | 实验编号 |
| result | 本实验成绩，可以为空，为空表示没有批改。 |
| memo | 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 |