

软件工程管理与经济课程项目

——虚拟实验仿真系统

软件设计规格说明书

SOFTWARE DESIGNS SPECIFICATION

小组成员：2051196 刘一飞

2052348 王杨乐

2050747 赵帅涛

2050865 黄彦铭

指导教师：黄杰

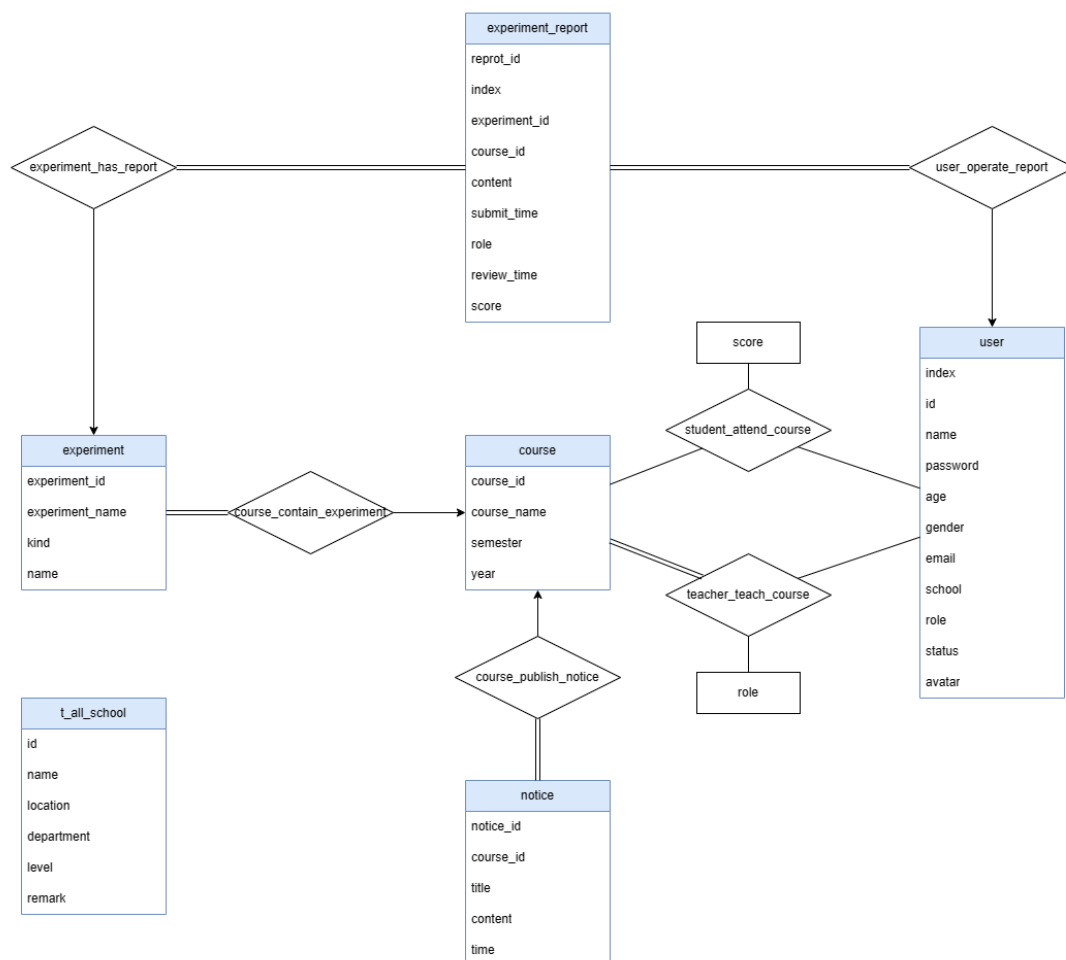
目 录

1 数据设计	2
1.1 数据库设计	2
1.1.1 逻辑设计	2
1.1.2 物理设计	2
2 架构设计	3
2.1 总体架构	3
2.1.1 前端架构设计	3
2.1.2 后端架构设计	3
2.2 架构上下文	4
3 接口设计	5
3.1 图形用户界面	5
3.2 内部接口	7
3.2.1 用户登录子系统	7
3.2.2 个人信息子系统	9
3.2.3 课程管理子系统	12
3.2.4 实验管理子系统	14
3.2.5 报告评阅子系统	24
3.2.6 授课管理子系统	28
3.3 外部接口	35
3.3.1 JavaMail 邮件发送	35
4 项目代码	36

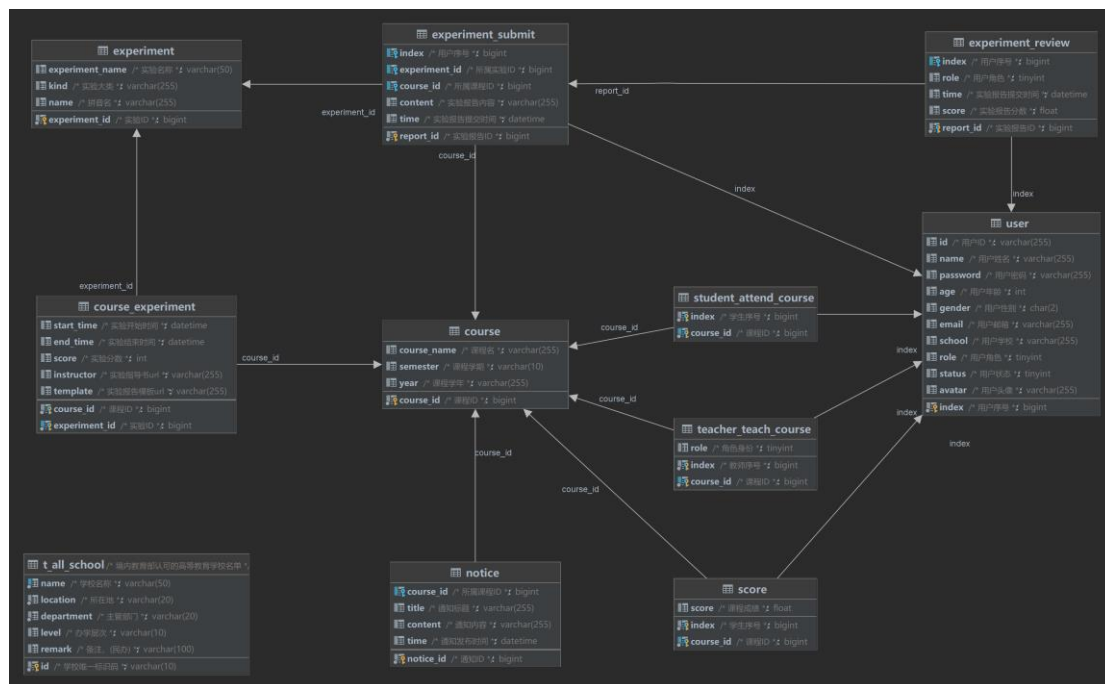
1 数据设计

1.1 数据库设计

1.1.1 逻辑设计



1.1.2 物理设计



2 架构设计

2.1 总体架构

2.1.1 前端架构设计

为满足系统的跨平台要求和提高易用性，前端借助浏览器以 web 页面的形式呈现。以 vue.js 作为主体开发框架；为达到系统界面的美观性，使用了 Element UI 组件库，借助 vue-router 进行路由导航；vuex 做前端存储管理；利用 Ajax 动态请求数据，更新页面，提升用户友好性，使用第三方库 Axios 对 Ajax 进行封装，降低开发难度。

vue 是一款用于构建用户界面的 JavaScript 框架，其基于标准 HTML，CSS 和 JavaScript，并提供了声明式，组件化的编程模型。借助其响应性特性，动态更新页面 DOM，做到页面渲染和内部数据的双向同步。利用 vue 提供的单页面文件支持，模块化前端页面元素，降低系统耦合性。

Ant design 将前端页面常用组件进行封装，提供了大量美观、易用的 UI 元素，借助该组件库，有效降低系统开发难度，提升系统观感。

vue-router 为单页面应用而生，可方便的做到嵌套路由映射，动态路由导航，通过与 vuex 配合使用，动态生成路由表，做到细致的权限管理。

2.1.2 后端架构设计

后端以 Java 作为开发语言，主要使用 spring boot 架构进行开发，主要考虑了以下几

个要点：

1. 数据访问层 (Data Access Layer)：Spring Boot 提供了强大的数据访问层支持，包括对 JDBC、ORM（如 Hibernate）和 NoSQL 数据库（如 MongoDB）的集成。它简化了数据访问的配置和操作，提供了事务管理和持久化等功能。
2. Web 层：Spring Boot 内置了 Spring MVC 框架，用于构建 Web 应用程序。它提供了处理请求和响应的控制器、视图解析器、拦截器等组件。同时，Spring Boot 还支持 RESTful 风格的 Web 服务开发，可以使用 Spring WebFlux 构建反应式 Web 应用。
3. 安全性 (Security)：Spring Boot 提供了对应用程序安全的支持。它包括认证 (Authentication) 和授权 (Authorization) 等功能，可以轻松集成常见的身份验证和授权机制，如基于角色的访问控制、OAuth 2.0 等。

2.2 架构上下文

1. 系统描述：
 - 该虚拟仿真实验平台旨在提供一种基于 Web 的虚拟实验环境，供用户进行各种实验和模拟操作。
 - 用户可以通过前端界面与后端系统进行交互，并获取实验结果和反馈信息。
2. 系统边界：
 - 前端界面：基于 Vue 框架搭建的单页应用 (SPA)，提供用户界面、实验控制和结果展示等功能。
 - 后端系统：基于 Spring Boot 框架构建的应用程序，处理前端请求、执行实验逻辑、与数据库交互等。
3. 前端组件：
 - 页面组件：包括实验列表、实验详情、实验控制、结果展示等页面，通过 Vue 组件实现。
 - 路由组件：通过 Vue Router 管理不同页面之间的导航和路由。
 - HTTP 请求组件：使用 Vue 的 HTTP 库或第三方库进行与后端 API 的通信。
4. 后端组件：
 - 控制器 (Controller)：负责接收前端请求、处理请求参数、调用相应的服务和管理数据的输入输出。
 - 服务层 (Service)：实现具体的实验逻辑、数据处理的业务规则等，处理来自控制器的请求。

- 数据访问层 (Data Access Layer): 与数据库进行交互, 执行数据的读写操作。
- 安全层 (Security Layer): 处理用户身份认证和权限控制, 确保实验数据和用户信息的安全性。
- 第三方服务集成: 如果需要, 与其他外部服务 (如身份验证服务、存储服务) 进行集成。

5. 依赖关系:

- 前端依赖后端: 前端通过 HTTP 请求调用后端提供的 API, 获取实验数据、发送控制指令等。
- 后端依赖前端: 后端提供 API 供前端调用, 接收和处理前端的请求。

6. 部署架构:

- 前端部署: 前端代码可以通过静态资源服务器或 CDN 进行部署, 供用户通过浏览器访问。
- 后端部署: 后端代码可以部署在应用服务器 (如 Tomcat) 上, 通过 HTTP 协议接收和处理前端请求。

7. 非功能需求:

- 性能: 设计优化后端的响应速度和吞吐量, 以确保实验平台的快速和流畅使用。
- 可扩展性: 采用水平扩展和面向对象的方式, 支持对实验平台的横向扩展和模块化拆分。
- 安全性: 实现身份验证和授权机制, 保护实验数据的机密性和完整性。

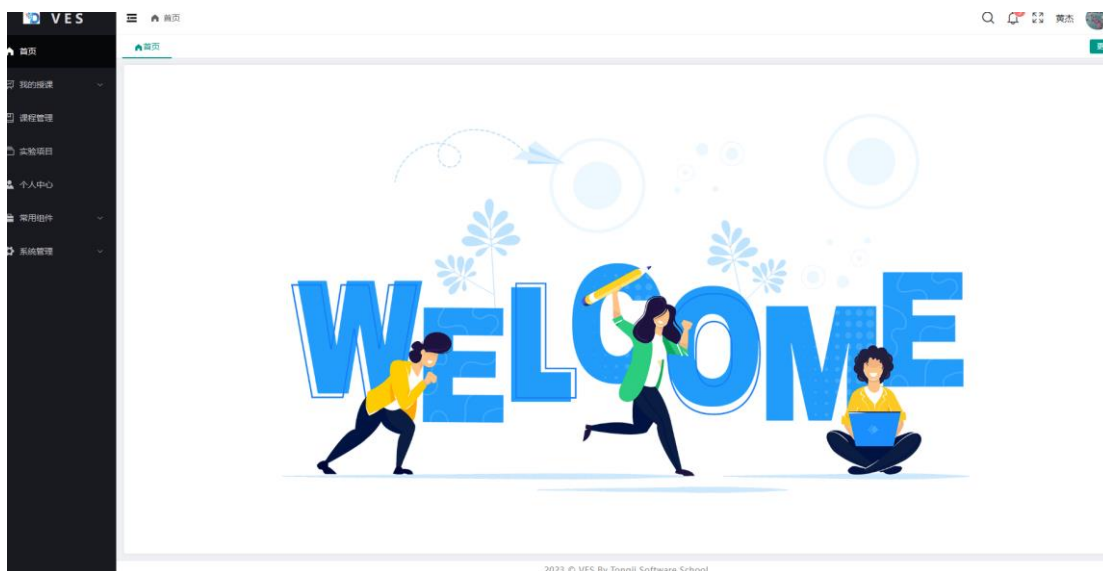
3 接口设计

3.1 图形用户界面

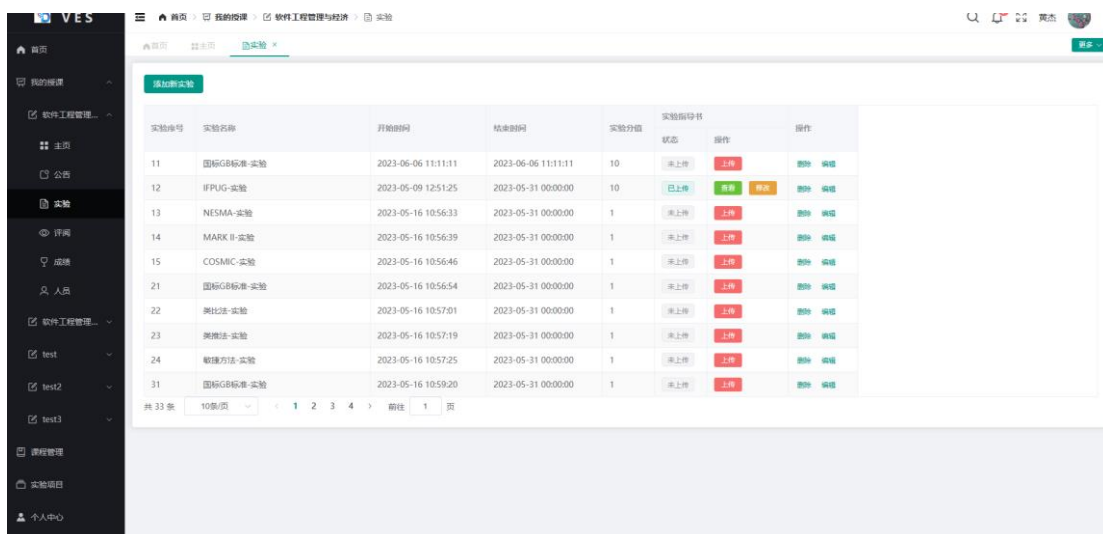
登陆界面:

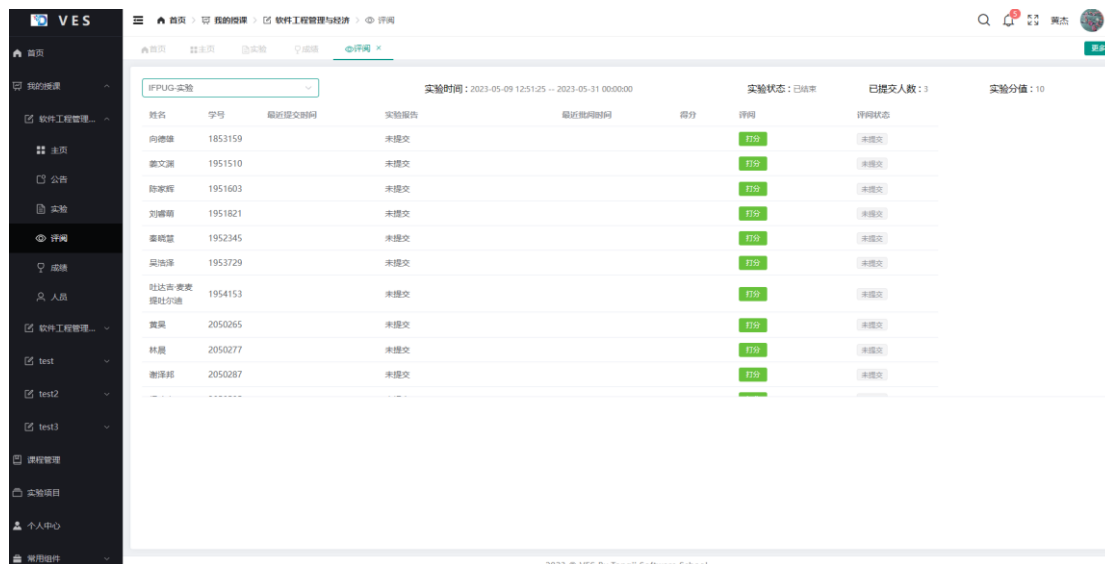


主要页面：



功能页面示例：





3.2 内部接口

3.2.1 用户登录子系统

3.2.1.1 用户账户激活

接口地址	/api/activate
请求方式	POST
请求数据类型	application/json
响应数据类型	application/json
接口描述	用于用户第一次登录系统时的账户激活

- 请求示例

```
JSON
{
  "code": "600162",
  "password": "123456",
  "username": "2051196@tongji.edu.cn"
}
```

- 请求参数

参数名称	参数说明	类型	备注
code	邮箱验证码	string	
password	账户密码	string	新设置的密码不能与旧密码相同
username	用户名	string	用户名即为对应邮箱

- 响应示例

```
JSON
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": "账户激活成功"
}
```

3.2.1.2 用户账户登录

接口地址	/api/login
请求方式	POST
请求数据类型	application/json
响应数据类型	application/json
接口描述	用于用户的账户登录

- 请求示例

```
JSON
{
  "username": "2051196@tongji.edu.cn",
  "school": "同济大学",
  "password": "3155002905"
}
```

- 请求参数

参数名称	参数说明	类型	备注
username	用户名	string	用户名即为对应邮箱
school	学校	string	
password	账户密码	string	新设置的密码不能与旧密码相同

- 响应示例

```
JSON
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": {
    "token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhdWQiOiI0NiIsImV4cCI6MTY4NDMwOTU1NH0.CHhLESSgeZtAzBnLGqloc1s9Xwud2QZ7AhjUfSRSCg0"
  }
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
token	身份验证令牌	string	身份令牌用于在后续的请求中验证用户身份

3.2.2 个人信息子系统

3.2.2.1 用户修改密码

接口地址	/change-password
请求方式	POST
请求数据类型	application/json
响应数据类型	application/json

接口描述	用于修改登录用户的密码
------	-------------

- 请求示例

JSON
/change-password?password=\${password}

- 响应示例

```
JavaScript
{
  "code" : 0,
  "data" : "用户修改密码成功",
  "success" : true,
  "msg" : null
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
data	修改响应提示	string	提示是否修改成功

3.2.2.2 用户修改头像

接口地址	/user/change-avatar
请求方式	POST
请求数据类型	multipart/form-data
响应数据类型	application/json
接口描述	用于修改登录用户的头像

- 请求示例

```
Shell
curl -X POST "http://127.0.0.1:8001/api/user/change-avatar" -H
"accept: */*" -H "Content-Type: multipart/form-data" -F
"file=@0IP-C.jpg;type=image/jpeg"
```

- 响应示例

```
JavaScript
{
  "code" : 0,
  "data" : {
    "fileUrl":"https://example.com/img.jpg"
  },
  "success" : true,
  "msg" : null
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
fileUrl	新头像的 URL	string	

3.2.2.3 用户修改个人信息

接口地址	/review/change-userinfo
请求方式	PUT
请求数据类型	application/json
响应数据类型	application/json
接口描述	用于修改登录用户的个人信息

- 请求示例

```
JSON
{
  "age": 0,
  "avatar": "string",
  "email": "string",
  "gender": "string",
  "name": "string"
}
```

- 响应示例

```
JSON
{
  "code" : 0,
  "data" : true,
  "success" : true,
  "msg" : null
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
data	修改响应提示	Boolean	提示是否修改成功

3.2.3 课程管理子系统

3.2.3.1 教师新增课程

接口地址	/api/addcourse
请求方式	POST
请求数据类型	application/json
响应数据类型	application/json
接口描述	新增一门课程

- 请求示例

```
JSON
{
  "courseName": "string",
  "semester": "string",
  "studentList": [
    {
      "姓名": "string",
      "学号": 0,
      "学校": "string",
      "性别": "string",
      "邮件": "string"
    }
  ]
}
```

```
    }
  ],
  "teacherList": [
    0
  ],
  "year": 0
}
```

- 响应示例

```
JSON
{
  "code": 0,
  "data": "string",
  "msg": "string",
  "success": true
}
```

3.2.3.2 教师获取所有课程

接口地址	/api/courses
请求方式	GET
请求数据类型	application/json
响应数据类型	application/json
接口描述	获取一名教师的所有课程

- 请求示例

```
JSON
Authorization: Bearer <token>
{
  curl -X GET "http://139.196.226.104:8001/api/courses?index=2"
  -H "accept: */*"
}
```

- 请求参数

参数名称	参数说明	数据类型	备注
------	------	------	----

index	教师 index	string	
-------	----------	--------	--

- 响应示例

```
JSON
{
  "code": 0,
  "data": [
    {
      "courseId": 0,
      "courseName": "string",
      "semester": "string",
      "year": 0
    }
  ],
  "msg": "string",
  "success": true
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
courseId	课程 Id	number	
courseName	课程名称	string	
semester	开设学期	string	
year	开设年份	number	

3.2.4 实验管理子系统

3.2.4.1 教师获取所有实验

接口地址	/api/allexperiment
请求方式	GET
请求数据类型	application/json

响应数据类型	application/json
接口描述	用于获取所有的实验信息

- 请求示例

```
JSON
Authorization: Bearer <token>
{}
```

- 响应示例

```
JSON
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": [
    {
      "experimentId": "11",
      "experimentName": "国标 GB 标准-实验",
      "kind": "软件规模度量",
      "name": "gb"
    },
    {
      "experimentId": "12",
      "experimentName": "IFPUG-实验",
      "kind": "软件规模度量",
      "name": "ifpug"
    },
    ...
  ]
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
experimentId	实验 Id	string	
experimentName	实验名称	string	

kind	实验大类	string	
name	实验名称缩写	string	

3.2.4.2 教师获取单个实验

接口地址	/api/experiment/{experimentId}
请求方式	GET
请求数据类型	application/json
响应数据类型	application/json
接口描述	用于根据实验 Id 获取单个实验的信息

- 请求示例

```
JSON
Authorization: Bearer <token>
{
  "experimentId": "11"
}
```

- 请求参数

参数名称	参数说明	类型	备注
experimentId	实验 Id	string	

- 响应示例

```
JSON
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": {
    "experimentId": "11",
    "experimentName": "国标 GB 标准-实验",
    "kind": "软件规模度量",
  }
}
```

```
    "name": "gb"
  }
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
experimentId	实验 Id	string	
experimentName	实验名称	string	
kind	实验大类	string	
name	实验名称缩写	string	

3.2.4.3 获取课程中的所有实验

接口地址	/api/experimentincourse
请求方式	GET
请求数据类型	application/json
响应数据类型	application/json
接口描述	用于根据课程 Id，获取所有在该课程中的实验

- 请求示例

```
JSON
Authorization: Bearer <token>
{
  "courseId": "42041301"
}
```

- 请求参数

参数名称	参数说明	类型	备注
------	------	----	----

courseId	课程 Id	string	
----------	-------	--------	--

- 响应示例

```
JSON
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": [
    {
      "experimentId": "12",
      "experimentName": "IFPUG-实验",
      "instructor": "https://virtual-experiment-system.oss-cn-shanghai.aliyuncs.com/09-05-2023/969fb6b73d9a4c6abf253cf3e9eeeab7/软件工程管理与经济-实验指导书-实验二_小型软件项目规模度量实验.pdf",
      "score": 10,
      "startTime": "2023-05-09T04:51:25.000+00:00",
      "endTime": "2023-05-30T16:00:00.000+00:00"
    }
  ]
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
experimentId	实验 Id	string	
experimentName	实验名称	string	
instructor	实验指导书	string	实验指导书的 url
score	实验分数	number	
startTime	实验开始时间	timestamp	
endTime	实验结束时间	timestamp	

3.2.4.4 获取不在课程中的所有实验

接口地址	/api/experimentnotincourse
请求方式	GET
请求数据类型	application/json
响应数据类型	application/json
接口描述	用于根据课程 Id，获取所有不在该课程中的实验

- 请求示例

```
JSON
Authorization: Bearer <token>
{
  "courseId": "42041301"
}
```

- 请求参数

参数名称	参数说明	类型	备注
courseId	课程 Id	string	

- 响应示例

```
JSON
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": [
    {
      "experimentId": "11",
      "experimentName": "国标 GB 标准-实验",
      "kind": "软件规模度量",
      "name": "gb"
    },
    {
      "experimentId": "13",
```

```
        "experimentName": "NESMA-实验",
        "kind": "软件规模度量",
        "name": "nesma"
    },
    ...
]
}
```

• 响应参数

参数名称	参数说明	数据类型	备注
experimentId	实验 Id	string	
experimentName	实验名称	string	
kind	实验大类	string	
name	实验名称缩写	string	

3.2.4.5 添加课程中的实验

接口地址	/api/add_experimentincourse
请求方式	POST
请求数据类型	application/json
响应数据类型	application/json
接口描述	用于根据课程 Id 和实验 Id，添加实验到课程中

• 请求示例

```
JSON
Authorization: Bearer <token>
{
    "courseId": "42041301",
    "experimentIdList": ["12", "13",...]
}
```

- 请求参数

参数名称	参数说明	类型	备注
courseId	课程 Id	string	
experimentIdList	实验 Id 数组	Array	数组中的实验 Id 都会加入到课程中

- 响应示例

```
JSON
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": "添加实验成功"
}
```

3.2.4.6 修改课程中的实验

接口地址	/api/modify_experimentincourse
请求方式	PUT
请求数据类型	application/json
响应数据类型	application/json
接口描述	用于根据课程 Id 和实验 Id，修改课程中的实验信息

- 请求示例

```
JSON
Authorization: Bearer <token>
{
  "courseId": "42041301",
  "experimentId": "11",
  "startTime": "yyyy-MM-dd hh:mm:ss",
  "endTime": "yyyy-MM-dd hh:mm:ss"
}
```

- 请求参数

参数名称	参数说明	类型	备注
courseId	课程 Id	string	
experimentId	实验 Id	string	
startTime	实验开始时间	timestamp	
endTime	实验结束时间	timestamp	

- 响应示例

```
JSON
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": "修改实验成功"
}
```

3.2.4.7 删除课程中的实验

接口地址	/api/delete_experimentincourse
请求方式	DELETE
请求数据类型	application/json
响应数据类型	application/json
接口描述	用于根据课程 Id 和实验 Id，删除课程中的指定实验

- 请求示例

```
JSON
Authorization: Bearer <token>
{
```

```
"courseId": "42041301",  
"experimentId": "12"  
}
```

- 请求参数

参数名称	参数说明	类型	备注
courseId	课程 Id	string	
experimentId	实验 Id	string	

- 响应示例

```
JSON  
{  
  "success": true,  
  "code": 0,  
  "msg": "success",  
  "data": "删除实验成功"  
}
```

3.2.4.8 上传实验指导书

接口地址	/api/upload_experiment_instructor
请求方式	POST
请求数据类型	multipart/form-data
响应数据类型	application/json
接口描述	用于根据课程 Id 和实验 Id，删除课程中的指定实验

- 请求示例

```
JSON  
Authorization: Bearer <token>  
{  
  "file": file  
  "courseId": "42041301",  
  "experimentId": "12"
```



```
}
```

- 请求参数

参数名称	参数说明	类型	备注
file	实验指导书文件	MultipartFile	
courseId	课程 Id	string	
experimentId	实验 Id	string	

- 响应示例

```
JSON
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": "上传实验指导书成功"
}
```

3.2.5 报告评阅子系统

3.2.5.1 获取实验详情

接口地址	/review/experimentlist
请求方式	GET
请求数据类型	application/json
响应数据类型	application/json
接口描述	获取当前课程的所有实验项目的详细信息

- 请求示例

```
JSON
/review/experimentlist?courseId=${courseId}
```

- 响应示例

```
JSON
{
  "code" : 0,
  "data" : {
    "endTime":"2023-06-12 12:00:00",
    "startTime":"2023-05-12 12:00:00",
    "experimentId":0,
    "experimentName":"IFPUG 实验",
    "score":20,
  },
  "success" : true,
  "msg" : null
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
startTime	实验开始时间	string	
endTime	实验结束时间	string	
experimentId	实验 ID	number	
experimentName	实验名称	string	
score	实验分值	number	

3.2.5.2 获取实验报告

接口地址	/review/reportlist
请求方式	GET
请求数据类型	application/json
响应数据类型	application/json

接口描述	获取当前课程的所有实验项目的详细信息
------	--------------------

- 请求示例

JSON

```
/review/experimentlist?courseId=${courseId}&experimentId=${experimentId}
```

- 响应示例

JSON

```
{
  "code" : 0,
  "data" : {
    "reportId":0,
    "reportUrl":"https://example.com/example.pdf",
    "reviewTime":"2023-06-12 12:00:00",
    "schoolNumber":"2050001",
    "score":0,
    "stuId":0,
    "stuName":"李华",
    "submitTime":"2023-06-12 12:00:00"
  },
  "success" : true,
  "msg" : null
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
reportId	实验报告标识	string	
reportUrl	实验报告访问URL	string	
reviewTime	评阅时间	string	
schoolNumber	学号	string	
score	报告得分	number	

stuld	用户唯一标识	number	
stuName	学生姓名	string	
submitTime	报告提交时间	string	

3.2.5.3 批阅实验报告

接口地址	/review/score
请求方式	POST
请求数据类型	application/json
响应数据类型	application/json
接口描述	获取当前课程的所有实验项目的详细信息

- 请求示例

```
JSON
{
  "reportId": 0,
  "reviewerId": 0,
  "reviewerRole": true,
  "score": 0
}
```

- 响应示例

```
JSON
{
  "code" : 0,
  "data" : true
  "success" : true,
  "msg" : null
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
------	------	------	----

data	是否批阅成功	Boolean	
------	--------	---------	--

3.2.6 授课管理子系统

3.2.6.1 获取课程公告

接口地址	/api/coursenotice
请求方式	GET
请求数据类型	application/json
响应数据类型	application/json
接口描述	根据课程 ID，获取该课程的公告

- 请求示例

HTTP

http://139.196.226.104:8001/api/coursenotice?courseId=1

- 响应示例

JSON

```
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": [
    {
      "noticeId": "7",
      "title": "test",
      "content": "<p><u><strong>test1</strong></u></p>",
      "time": "2023-05-07 00:00:00"
    }
  ]
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
------	------	------	----

noticeId	公告 Id	string	
title	公告标题	string	
content	公告内容	string	
time	公告发布时间	timestamp	

3.2.6.2 删除课程公告

接口地址	/api/coursenotice
请求方式	DELETE
请求数据类型	application/json
响应数据类型	application/json
接口描述	根据课程 ID，获取该课程的公告

- 请求示例

HTTP
http://139.196.226.104:8001/api/coursenotice?courseId=1

- 响应示例

```
JSON
{
  "code": 0,
  "data": "string",
  "msg": "string",
  "success": true
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
noticeId	公告 Id	string	

title	公告标题	string	
content	公告内容	string	
time	公告发布时间	timestamp	

3.2.6.3 新增课程公告

接口地址	/api/coursenoticeadd
请求方式	POST
请求数据类型	application/json
响应数据类型	application/json
接口描述	根据课程 ID，获取该课程的公告

- 请求示例

HTTP
http://139.196.226.104:8001/api/coursenotice?courseId=1

- 请求参数

参数名称	参数说明	数据类型	备注
courseId	课程 Id	number	
noticeId	公告 Id	number	
publishTime	公告发布时间	timestamp	
content	公告内容	string	
noticeTitle	公告标题	string	

- 响应示例

```
JSON
{
  "code": 0,
  "data": {
    "content": "string",
    "courseId": 0,
    "noticeId": 0,
    "time": {
      "date": 0,
      "day": 0,
      "hours": 0,
      "minutes": 0,
      "month": 0,
      "nanos": 0,
      "seconds": 0,
      "time": 0,
      "timezoneOffset": 0,
      "year": 0
    },
    "title": "string"
  },
  "msg": "string",
  "success": true
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
noticeId	公告 Id	string	
courseId	课程 Id	string	
title	公告标题	string	
content	公告内容	string	
time	公告发布时间	timestamp	

3.2.6.4 获取课程名单

接口地址	/api/coursestudent
请求方式	GET
请求数据类型	application/json
响应数据类型	application/json
接口描述	根据课程 ID，获取该课程的公告

- 请求示例

```
HTTP
curl -X GET
"http://139.196.226.104:8001/api/coursestudent?courseId=1" -H
"accept: */*"
```

- 响应示例

```
JSON
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": [
    {
      "stuId": "1853159",
      "school": "同济大学",
      "gender": "男",
      "email": "1853159@tongji.edu.cn",
      "name": "向德雄"
    },
    {
      "stuId": "1951510",
      "school": "同济大学",
      "gender": "男",
      "email": "1951510@tongji.edu.cn",
      "name": "姜文渊"
    },
    ....
  ],
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
stuld	学生 Id	string	
school	学校	string	
gender	性别	string	
email	邮件	string	
name	姓名	string	

3.2.6.5 获取选课学生列表

接口地址	/api/coursestudent
请求方式	GET
请求数据类型	application/json
响应数据类型	application/json
接口描述	根据课程 ID，获取该课程的公告

- 请求示例

```
HTTP
curl -X GET
"http://139.196.226.104:8001/api/coursefreestudent?courseId=42041301" -H "accept: */*"
```

- 响应示例

```
JSON
{
  "success": true,
  "code": 0,
  "msg": "success",
  "data": [
```

```
{
  "stuId": "1853159",
  "school": "同济大学",
  "gender": "男",
  "email": "1853159@tongji.edu.cn",
  "name": "向德雄"
},
{
  "stuId": "1951510",
  "school": "同济大学",
  "gender": "男",
  "email": "1951510@tongji.edu.cn",
  "name": "姜文渊"
},
....
],
}
```

- 响应参数

参数名称	参数说明	数据类型	备注
stuld	学生 Id	string	
school	学校	string	
gender	性别	string	
email	邮件	string	
name	姓名	string	

3.2.6.6 新增学生

接口地址	/api/coursestudentadd
请求方式	POST
请求数据类型	application/json

响应数据类型	application/json
接口描述	根据课程 ID，获取该课程的公告

- 请求示例

```
HTTP
POST http://localhost:8001/api/coursestudentadd
```

- 请求参数

参数名称	参数说明	数据类型	备注
courseId	课程 Id	number	
stuld	学生 Id	string	
name	学生姓名	string	
school	学校	string	

- 响应示例

```
JSON
{
  "code": 0,
  "data": "string",
  "msg": "string",
  "success": true
}
```

3.3 外部接口

3.3.1 JavaMail 邮件发送

通过调用 JavaMail API 实现用户在登录注册时需要完成的邮件验证功能，使用 SMTP 协议将邮件发送给对应用户。

4 项目代码

前端仓库: <https://github.com/PrettyMagnolia/VSE-Back-Manage-System-Front>

后端仓库: <https://github.com/PrettyMagnolia/VSE-Back-Manage-System-Back>