

橘子社

# 项目管理系统

# 系统设计说明书

文档状态： 草稿[ ] 正式发布[ ] 修订中[ ]	组织名称	橘子社
	项目名称	项目管理系统
	文件标识	SJ10001
	当前版本	1.0
	作者	王鸿博
	完成日期	2020-12-28

[illegible]

# 第一部分 概述

## 1. 文档说明

本文档适用于项目开发人员观看，根据文档进行开发。同时可用于实时审查与修改。采取 OOA&D 方法设计。

## 2. 系统需求概述

本项目目的是开发一个基于 B/S 的项目管理系统，满足项目管理需求。适用于学校实验项目，非实际商用化，功能会有差异。主要包含以下功能：

- ◎项目文档管理
- ◎项目沟通管理
- ◎项目进度监控
- ◎项目干系人管理

## 3. 系统总体结构

主要采用 MVC 模式和 B/S 风格，前后端通过网络通信的形式进行交互，后端基于 SpringBoot 框架进行设计开发。

## 4. 开发环境

在 windows10 平台，统一使用 IntelliJ Idea 开发工具，主要使用 Java 语言开发，JDK 版本 11。

## 5. 开发管理

使用 Github 进行项目开发文件管理

# 第二部分 数据库设计

## 1. 概要

选用 Mysql 数据库，版本为 mysql8.0

## 2. 数据表总览

表名	功能说明
Object	操作控制器
Operatioon	操作方法
PermissionConfig	许可配置
RolePowerAssignment	角色分配的许可
Role	用户角色
User	用户
Task	任务
FileAssess	文件评价
FileInfo	文件信息
FileContent	文件内容
Student	用户的子类，学生
StudentAssess	老师对学生的评价
Communication	聊天消息
Team	小组
GroupAssess	小组评价
ProjectCompletion	项目进度记录
Milestone	项目里程碑
ProjectAssignment	项目分配到的小组
Project	项目

## 3. 数据表设计

数据库名: pmcore		数据表名: Object			
存储组织: ObjectName 作为业务主键		字段数: 1	主键: ObjectName		
		数据量: 少	外键:		
字段组成: ObjectName : varchar					
简要说明: 以 springboot 的 controller 层的各个方法作为一个条目记录					
修改记录:		编写	王鸿博	日期	2020.12.28
		审核		日期	

数据库名: pmcore		数据表名: Operation	
存储组织: OperationDescription 作为业务主键	字段数: 1	主键: OperationDescription	
	数据量: 少	外键:	
字段组成: OperationDescription : varchar			

简要说明: 以网络通信 http 协议的几种方法(GET,POST 等)作为记录				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore		数据表名: PermissionConfig		
存储组织: ConfigID 代理主键, 自增	字段数: 4	主键: ConfigID		
	数据量: 较少	外键: ObjectName OperationDescription		
字段组成: ConfigID : int ObjectName : varchar OperationDescription : varchar Permission : int – 默认空				
简要说明: 以操作的函数名和方法类型确定许可信息				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore		数据表名: RolePowerAssignment		
存储组织: AssignmentID 代理主键, 自增。相当于 RoleID 和 ConfigID 组合主键。	字段数: 3	主键: AssignmentID		
	数据量: 中等	外键: RoleID ConfigID		
字段组成: AssignmentID : int RoleID : int ConfigID : int				
简要说明: 确定角色与权限配置之间的分配				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore	数据表名: Role
存储组织: RoleID 代理主键, 自增	字段数: 2 主键: RoleID 数据量: 少 外键:
字段组成: RoleID : int RoleName : varchar – 默认空	
简要说明: 记录所有项目角色	

修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore		数据表名: User		
存储组织: UserID 作为代理主键, 自增	字段数: 8	主键: UserID		
	数据量: 多	外键: RoleID		
字段组成: UserID : int RoleID : int UserName : varchar – 默认空 UserPassword : varchar – 默认空 Tel : bigint – 默认空 EMail : varchar – 默认空 UserIconLocation : varchar – 默认空 Sex : varchar – 默认空				
简要说明: 记录用户账号 (ID) , 以及其他信息。包含学生和老师, 但老师不单独分类				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore		数据表名: Task		
存储组织: TaskID 代理主键, 自增, TaskPublisherID 为 User 表外键	字段数: 8	主键: TaskID		
	数据量: 多	外键: TaskPublisherID		
字段组成: TaskID : int TaskName : varchar - 默认空 TaskDeadline : date – 默认空 TaskHandlerID : int – 默认空 TaskFinishType : int – 默认空 TaskStartTime : date -默认空 TaskPublisherID : int TaskDescription : varchar – 默认空				
简要说明: 记录所有发布的任务, 不一定被接手, 因此 TaskHandler 不作为 User 外键				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore	数据表名: Student			
存储组织: 作为 User 的子类	字段数: 4	主键: StudentUserID		
	数据量: 多	外键: StudentUserID		

		TeamID		
字段组成:				
StudentUserID : int				
Class : varchar – 默认空				
StudentID : bigint – 默认空				
TeamID : int				
简要说明:				
记录学生用户，继承自 User。StudentID 是学号，注意区分				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore		数据表名: StudentAssess		
存储组织: StudentUserID 作为代理主键, 同时是 User 外键	字段数: 4	主键: StudentUserID		
	数据量: 多	外键: StudentUserID AssessorID		
字段组成: StudentUserID : int AssessorID : int Assess : varchar – 默认空 Score : int – 默认空				
简要说明: 记录老师对学生的评价/评分, AssessorID 为 User 外键, 为评价人				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore		数据表名: Team		
存储组织: TeamID 作为代理主键, 自增	字段数: 3	主键: TeamID		
	数据量: 较少	外键:		
字段组成: TeamID : int TeamName : varchar – 默认空 TeamDescription : varchar – 默认空				
简要说明: 记录所有队伍, 别名为 Group				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore	数据表名: Communication	
存储组织:	字段数: 5	主键: CommunicationID
CommunicationID 代理主键, 自增	数据量: 多	外键: UserID, TeamID

字段组成： CommunicationID : int UserID : int TeamID : int FileID : int – 默认空 Context : varchar – 默认空				
简要说明： 记录所有讨论。有可能是单纯的讨论，也有可能是针对某一个项目文档的讨论，因此引入 FileID，FileID 属于 FileInfo 表，但不作为外键				
修改记录：	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore		数据表名: GroupAssess		
存储组织: TeamID 作为代理主键，同时是外键	字段数: 4	主键: TeamID		
	数据量: 较少	外键: TeamID AssessorID		
字段组成: TeamID : int AssessorID : int TeamAssess : varchar – 默认空 Score : int – 默认空				
简要说明: 记录老师对小组的评分。				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore		数据表名: Project		
存储组织: ProjectID 代理主键, 自增	字段数: 3	主键: ProjectID		
	数据量: 少	外键:		
字段组成: ProjectID : int ProjectName : varchar – 默认空 ProjectDescription : varchar – 默认空				
简要说明: 记录所有发布的项目。包含项目名和描述				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore		数据表名: ProjectAssignment		
存储组织：	字段数: 4	主键: TeamID		

TeamID 作为代理主键，同时是引自 Team 表的外键	数据量：较少	外键：TeamID ProjectID		
字段组成： TeamID : int ProjectID : int ProjectStartTime : date – 非空 ProjectDeadline : date – 默认空				
简要说明： 记录小组与项目的分配情况，以 TeamID 作为主键。同时包含项目启动和完工日期				
修改记录：	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名：pmcore		数据表名：FileInfo		
存储组织： FileID 代理主键，自增	字段数：7	主键：FileID		
	数据量：多	外键：StudentUserID TeamID		
字段组成： FileID : int TeamID : int StudentUserID : int FileRealName : varchar – 默认空 FileType : varchar – 默认空 FileLocation : varchar – 默认空 UploadTime : date – 默认空				
简要说明： 记录所有文件信息。以及所属项目，和上传人 ID。FileRealName 指示文件名，FileType 指示文件后缀，FileLocation 指示文件所在位置				
修改记录：	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名：pmcore		数据表名：FileContent			
存储组织： FileID 代理主键，同时是引自 FileInfo 表的外键	字段数：2		主键：FileID		
	数据量：多		外键：FileID		
字段组成： FileID : int FileContent : varchar – 默认空					
简要说明： 记录存储在数据库中（FileLocation）的文件的实际内容。					
修改记录：		编写	王鸿博	日期	2020.12.28



	审核		日期	
--	----	--	----	--

数据库名: pmcore		数据表名: FileAssess		
存储组织: FileID 代理主键, 引自 FileInfo 表。 AssessorID 引自 User 表	字段数: 4	主键: FileID		
	数据量: 多	外键: FileID AssessorID		
字段组成: FileID : int AssessorID : int FileAssess : varchar – 默认空 AssessTime : date				
简要说明: 记录老师对项目文件的评价。				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

数据库名: pmcore		数据表名: ProjectCompletion		
存储组织: CompletionID 代理主键, 自增	字段数: 4	主键: CompletionID		
	数据量: 少	外键: FileID TeamID MilestoneID		
字段组成: CompletionID : int FileID : int TeamID : int MilestoneID : int				
简要说明: 记录项目进度, 项目进度以对应文件的上传作为标志, 同时指示对应了哪一个里程碑				
修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

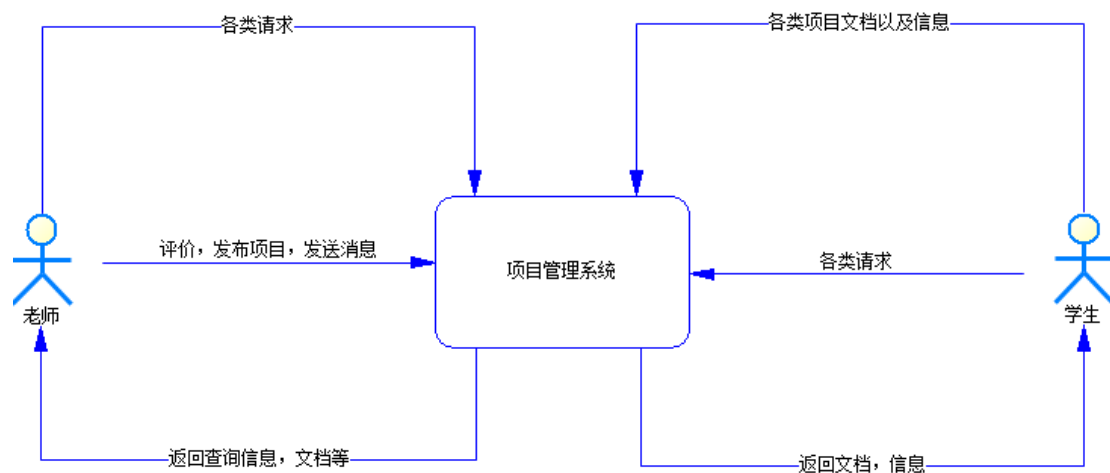
数据库名: pmcore		数据表名: Milestone	
存储组织: MilestoneID 代理主键, 自增	字段数: 3	主键: MilestoneID	
	数据量: 较少	外键:	
字段组成: MilestoneID : int MilestoneName : varchar – 默认空 MilestoneDescription : varchar – 默认空			
简要说明: 项目里程碑字典。记录项目各个里程碑及其描述			

修改记录:	编写	王鸿博	日期	2020.12.28
	审核		日期	

## 第三部分 系统功能设计

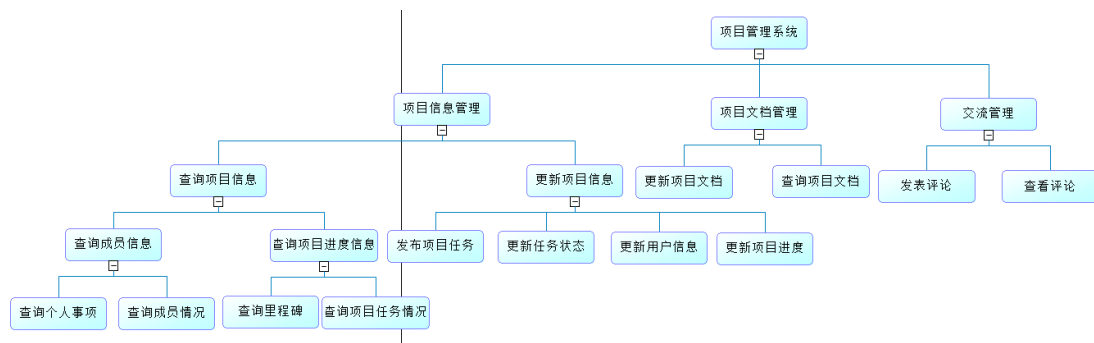
### 1. 上下文图

版本	ver1.0
修改人	王鸿博
修改日期	2020.11.28
备注	初步设计

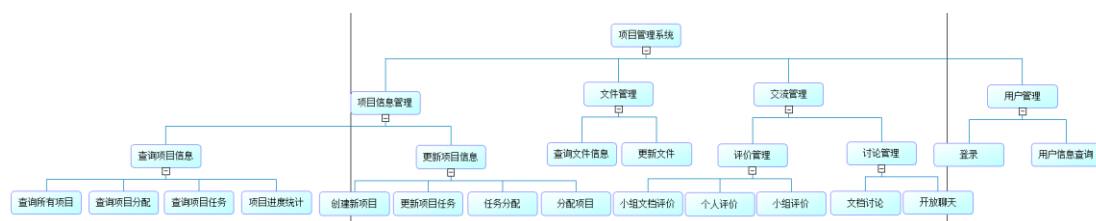


### 2. 分解图

版本	ver1.0
修改人	王鸿博
修改日期	2020.11.28
备注	初步设计

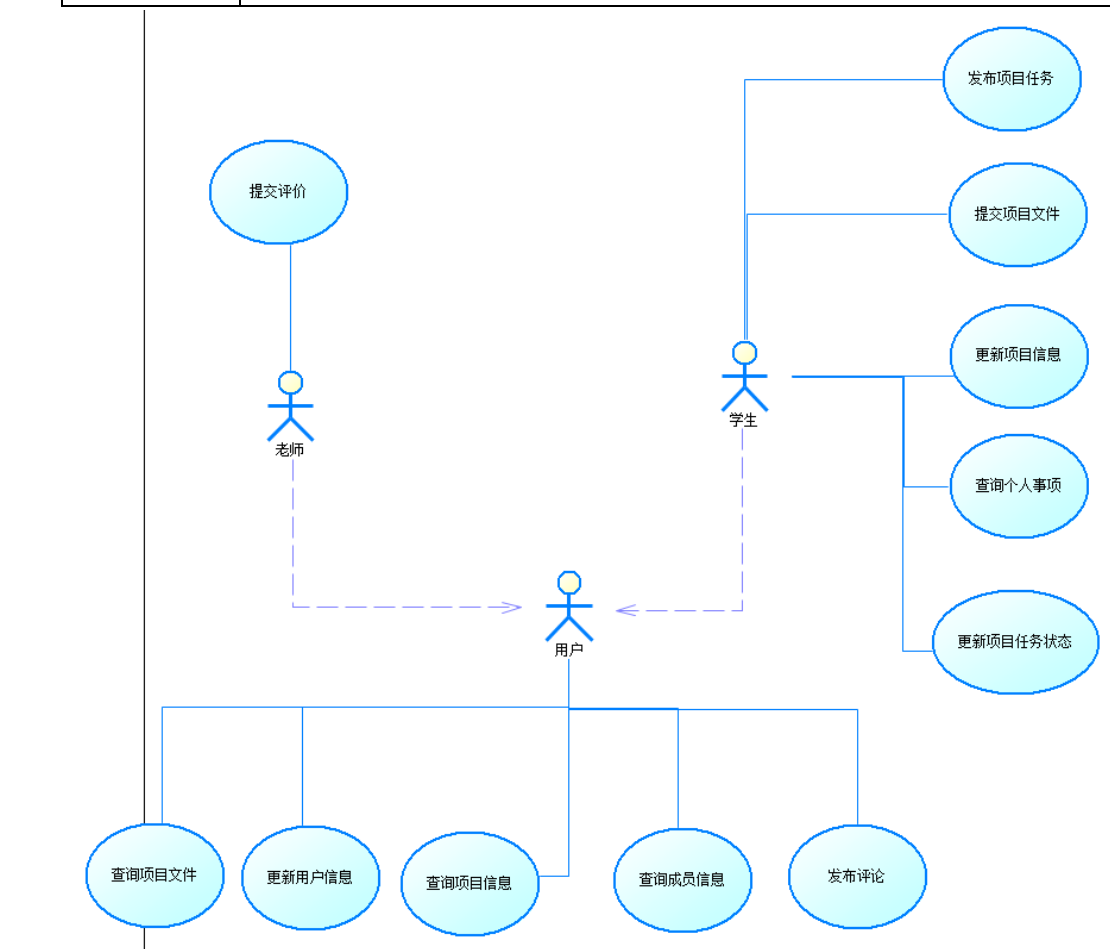


版本	ver2.0
修改人	王鸿博
修改日期	2020.12.29
备注	较第一版有大幅修改



### 3. 用例图

版本	ver1.0
修改人	王鸿博
修改日期	2020.11.28
备注	初步设计





用例名：更新项目任务		编号：UC03
用例概述	添加、删除以及修改任务	
主参与者	学生（组长）	
次参与者	学生（组员）	
前置条件	正确登入，且为组长发出请求	
后置条件	检查账号真实性、目标任务是否存在以及权限	
执行结果	返回成功信息或报告问题	

用例名：更新项目文件		编号：UC04
用例概述	上传、删除、覆盖项目文件	
主参与者	学生	
次参与者		
前置条件	正确登入且确实地选择了文件	
后置条件	检查权限和账号，检查文件真实性	
执行结果	返回成功或报告问题	

用例名：任务分配		编号：UC05
用例概述	修改任务，指定任务执行人	
主参与者	学生	
次参与者	学生	
前置条件	正确登入，且选择了正确的任务	
后置条件	检查账号和权限，检查任务状态	
执行结果	返回成功或报告问题	

用例名：分配项目		编号：UC06
用例概述	小组选择要实行的项目	
主参与者	学生（组长）	
次参与者		
前置条件	正确登入，且正确地选择了项目	
后置条件	检查账户和权限，检查项目状态	
执行结果	返回成功或报告错误，完成项目分配	

用例名：项目进度统计		编号：UC07
用例概述	统计项目总体进度	
主参与者	用户	
次参与者		
前置条件	正确登入，选择正确的项目	
后置条件	检查账户和权限，检查目标项目信息	
执行结果	返回项目进度信息或报告错误	

用例名：查询项目文件		编号：UC08
用例概述	可以查询所有文件或者是查询有特定作用的文件（如：项目章程）	
主参与者	用户	
次参与者		
前置条件	正确登入，选择正确的文件，学生只能选择查询自己组的项目文件	
后置条件	检查账户和权限，检查文件信息和用户信息	
执行结果	返回文件列表或下载链接，或报告错误	

用例名：用户信息查询		编号：UC09
用例概述	查询个人所属项目，目前任务	
主参与者	学生	
次参与者		
前置条件	正确登入	
后置条件	检查账户，查询信息	
执行结果	返回所需结果或报告错误	

用例名：查询项目任务		编号：UC10
用例概述	查询小组当前的任务	
主参与者	用户	
次参与者		
前置条件	正确登入，学生只能选择查询自己小组	
后置条件	检查账户，检查目标小组和用户信息	
执行结果	返回任务列表或报告错误	

用例名：查询项目分配		编号：UC11
用例概述	查询选择某一项目的所有组	
主参与者	用户	
次参与者		
前置条件	正确登入，正确选择项目	
后置条件	检查账户和权限，检查目标项目信息	
执行结果	返回小组列表或报告错误	

用例名：查看讨论		编号：UC12
用例概述	查看讨论	
主参与者	用户	
次参与者		

前置条件	正确登入，选择正确地查询范围。学生只能看到属于自己小组的消息讨论
后置条件	检查账户和权限，查询目标记录
执行结果	返回对应的讨论记录或报告错误

用例名：开放讨论 编号：UC13	
用例概述	发表针对整个小组的讨论
主参与者	用户
次参与者	
前置条件	正确登入
后置条件	检查账户。
执行结果	显示消息，记录消息。或是报告错误

用例名：文档讨论 编号：UC14	
用例概述	针对某一项目文档发起的讨论
主参与者	用户
次参与者	
前置条件	正确登入，正确地选择了文件
后置条件	检查账户，检查目标文件信息
执行结果	显示消息，记录消息。或是报告错误

用例名：发表小组评价 编号：UC15	
用例概述	老师对小组进行打分和总结
主参与者	用户（老师）
次参与者	学生
前置条件	正确登入，在老师的界面发出请求，正确选择小组
后置条件	检查账户。检查目标小组信息
执行结果	记录打分和评价，返回成功或报告问题

用例名：发表个人评价 编号：UC16	
用例概述	老师对个人表现进行打分和评价
主参与者	用户（老师）
次参与者	学生
前置条件	正确登入，在老师界面发起请求，正确选择学生
后置条件	检查账户，检查目标学生信息。
执行结果	记录打分和评价，返回成功或报告问题

用例名：发表文档评价 编号：UC17	
用例概述	老师发表对小组的项目文档的评价

主参与者	用户（老师）
次参与者	
前置条件	正确登入，在老师界面发起请求，正确选择文档
后置条件	检查账户，检查目标文件信息
执行结果	记录打分和评价，返回成功或报告问题

用例名：创建项目	编号：UC18
用例概述	创建新项目
主参与者	用户（老师）
次参与者	
前置条件	正确登入，在老师界面发起请求
后置条件	检查账户。检查项目是否冲突
执行结果	返回成功，记录项目。或是报告错误

## 第四部分 代码结构设计

### 1. Http 接口设计

版本	ver1.0
修改人	钟臻
修改日期	2020.11.27
备注	<p>初版设计，比较粗糙</p> <p>说明： 暂时的设计，应该还没弄完，以及有待优化完善。url 前面是服务器域名，先不管。method 方法类型； param 传递的参数； 返回值类型 application/json 自己先了解； 这些值要参照数据库设计（现在还没完全做完</p> <p>要求： 看懂下面的内容，不懂就问，然后可以提一下优化方案。如上所说这是未完成的，比如像 login 里的返回值 finish，如果登陆成功，其内容是什么，如果失败又是什么，这些还没有完全确定（举例：成功返回“yes”，失败返回“no 100（错误码） adajshd（错误描述）”。</p>

URL: /login

method: POST

param: UserID + UserPassword

return application/json

```
{
  "finish" : string,
  "loginType" : string,
```



```
"TeamID" : int,  
"TeamName" : string,  
"RoleName" : string  
}
```

URL: /file/\$fileID //文件下载

method: GET

param: 路由中就有的\$fileID 以及用 cookie 传递的 UserID、UserPassword

return 一个文件, 文件类型可以取: 图片、Office 全家桶的各种格式的文件、压缩包

//Files PermissionConfig

URL: /file //查看本组文件信息

method: GET

param: UserID、UserPassword

return application/json

```
{  
  "finish" : string,  
  "files" : [  
    {  
      "fileID" : int,  
      "fileRealName" : string,  
      "fileType" : string,  
      "uploadTime" : string  
      "StudentUserID" : int  
    }  
  ]  
}  
//Files PermissionConfig
```

URL: /file

method: POST //上传文件

param: 一个文件、UserID、UserPassword

return application/json

```
{
```

```
    "finish" : string
}
//Files    PermissionConfig

URL: /file
method: PUT    //更改文件
param: 文件 ID、一个新文件、UserID、UserPassword
return application/json
{
    "finish" : string
}
//Files    PermissionConfig
```

```
URL: /file    //删除文件
method: DELETE
param: 文件 ID、UserID、UserPassword
return application/json
{
    "finish" : string
}
//Files    PermissionConfig
```

```
URL: /task    //查看自己任务信息
method: GET
param: UserID、UserPassword
return application/json
{
    "finish" : string,
    "tasks" : [
        {
            "taskID" : int,
            "taskName" : string,
            "taskFinishType" : string,
            "taskStartTime" : string,
```

```

        "taskDeadline" : string,
        "taskHandlerID" : int,
        "taskHandlerName" : string,
        "taskPublisherID" : int,
        "taskPublisherName" : string
        "taskDescription" :string
    }
]
}
//Task      Student      PermissionConfig

```

URL: /task //发布任务

method: POST

param: UserID、UserPassword、还要有一个 task

task 的格式: {

```

    "taskID" : int,
    "taskName" : string,
    "taskFinishType" : string,
    "taskStartTime" : string,
    "taskDeadline" : string,
    "taskHandlerID" : int,
    "taskPublisherID" : int
    "taskDescription" :string

```

}

return application/json

```

{
    "finish" : string
}

```

//Task PermissionConfig

URL: /task //更改任务

method: PUT

param: TeamID(原), UserID、UserPassword、还要有一个新 task

```
task 的格式:{
    "taskID" : int,
    "taskName" : string,
    "taskDeadline" : data,
    "taskFinishType" : string,
    "taskDeadline" : data,
    "taskHandlerID" : int
    "taskPublisher":string
    "taskDescription" :string
}
return application/json
{
    "finish" : string
}

//Task      Student      PermissionConfig

URL: /task      //删除任务
method: DELETE
param: UserID、UserPassword、TaskID
return application/json
{
    "finish" : string
}

//Task      PermissionConfig

URL:/milestone      //查看里程碑信息
method:GET
param:UserID,UserPassword
return application/json
{
    "fileID": int,
    "milestoneName" : string,
```

```
    "milestoneDescription" : string
    "Files.UploadTime":datetime
}
```

```
//Milestone    PermissionConfig
```

```
URL:/milestone    //新增里程碑信息
method:POST
param:UserID,UserPassword,FileID,一个 milestone
return application/json
{
    "finish" : string
}
```

```
//Milestone    PermissionConfig
```

```
URL:/milestone    //更改里程碑信息
method:PUT
param:UserID,UserPassword,MilestoneID(原),FilesID,一个新的 milestone
return application/json
{
    "finish" : string
}
```

```
//Milestone    PermissionConfig
```

```
URL:/milestone    //删除里程碑信息
method:DELETE
param:UserID,UserPassword,MilestoneID
return application/json
{
    "finish" : string
}
```

//Milestone      PermissionConfig

URL:/comunication      //查看评价  
method:GET  
param:TeamID,FileID,UserID,UserPassword,  
return {  
    "context": char  
}

//Student          Group          Files          Comunication

URL:/comunication      //发布评价(fileID 为空时, 为老师和学生的交流)  
method:POST  
param: TeamID, fileID(nullable), context, UserID,UserPassword  
return {  
    "finish" : string  
}

//PermissionConfig      Files

URL:/comunication      //更改评价  
method:PUT  
param: ComunicationID(原),新的 comunication, UserID,UserPassword  
return {  
    "finish" : string  
}

//PermissionConfig      Files

URL:/comunication      //删除(撤回)评价  
method:DELETE  
param: Comunication,UserID,UserPassword

```
return {  
    "finish" : string  
}
```

```
//RolePowerAssignment      Files
```

//////// 这是老师评价体系, 只有老师能评价, 评价输入为 userID+userPassword+评价的事物或人, 后端检查是否是老师

//////// 而 GET 是查询功能, 输入是查询的事物, 返回是查询结果。查询时需要先查找 PermissionConfig 中 Permission 值是否为 1

```
URL:/studentAssess      //查看学生评分
```

```
method:GET
```

```
param:StudentID,UserID,UserPassword
```

```
return{  
    "StudentUserID":int  
    "StudentID":int  
    "UserName":String  
    "Score":int  
}
```

```
//PermissionConfig      Student      User      StudentAssess
```

```
URL:/studentAssess      //老师评分
```

```
method:POST
```

```
param:StudentID,UserID,UserPassword,一个 StudentAssess 表
```

```
return{  
    "finish" : string  
}
```

```
//PermissionConfig      StudentAssess
```

```
URL:/groupAccess
```

```
method:GET
```

```
param:TeamID,UserID,UserPassword
```

```
return{  
    "TeamID":int
```

```
    "Score":int
}

//PermissionConfig    GroupAssess    Group

URL:/groupAccess
method:POST
param:TeamID,UserID,UserPassword, 一个 GroupAssess 表
return{
    "finish":String
}

//PermissionConfig    GroupAssess

URL:/fileAccess
method:GET
param:FileID,UserID,UserPassword
return{
    "FileID":int
    "FileRealName":String
    "FileAssess":char
}

//PermissionConfig    Files

URL:/fileAccess
method:POST
param:FileID,UserID,UserPassword,一个 FileAssess 表
return{
    "finish":String
}

//PermissionConfig    Files

URL:/project
```



method:GET

param:projectID,UserID,UserPassword

return {

    "projectID" : int,

    "projectName":string

    "projectDescription" : string

}

//Project          PermissionConfig

URL:/project

method:POST

param: userID,userPassword,projectID,projectName,Description

return {

    "finish" : string

}

//Project          PermissionConfig

URL:/projectAssignment/\$projectID

method:GET

param:projectID,UserID,UserPassword

return {

    "teamID" : int,

    "projectStartTime" : string,

    "projectDeadLine" : string

}

//ProjectAssignment      Project

版本	ver2.0
修改人	王鸿博
修改日期	2020.12.25
备注	在第一版的基础上优化和修改，改变表现形式

地址头	后接地址/参数	接口类型	输入参数	返回结果	描述
/login	/	post	UserID - 用户 ID UserPassword - 用户密码	登陆结果, 小组 id, 小组名, 角色名	登录
/file	/fileID	get	response - 用于传输文件的 HTTPServletResponse fileID - 需要获取的文件	文件, 结果信息	根据文件 id 查询文件
	/	post	file - 文件本体 userID - 用户名, 用于 RBAC userPassword - 用户名, 用于 RBAC	结果信息	上传文件
	/	put	fileID - 被更新的文档的 ID file - 文件本体 userID - 用户名, 用于 RBAC userPassword - 用户名, 用于 RBAC	结果信息	更新文件
	/	delete	FileID - 被删除的文件的 ID UserID - 用户名, 用于 RBAC UserPassword - 用户密码, 用于 RBAC	结果信息	删除文件
/files	?{TeamID}	get	TeamID - 队伍 ID	文件信息列表, 完成 状态	获取整个队伍的文档列表
/task	?{UserID}	get	UserID 用户名	任务列表	寻找与用户有关的 Task
	/	post	userID - 账号 userPassword - 密码 taskName - Task 名 taskFinishType - Task 初始完成情况 taskDeadline - Task 截止时间 taskStartTime - Task 开始时间 taskHandlerID - Task 执行者 taskPublisherID - Task 发布者 taskDescription - Task 解释	结果信息	创建新 Task

	/	put	userID - 用户名 userPassword - 用户密码 taskID - 被修改的 Task 的 ID taskName - 被修改后的 Task 的 Name taskFinishType - 被修改后的 Task 的 FinishType taskDeadline - 被修改后的 Task 的 DeadLine taskStartTime - 被修改后的 Task 的 StartTime taskHandlerID - 被修改后的 Task 的 HandlerID taskPublisherID - 被修改后的 PublisherID taskDescription - 被修改后的 Task 的 Description teamID - 被修改后的 Task 的 teamID	结果信息	更改 Task
	/	delete	UserID - 用户名 UserPassword - 密码 taskID - 被删除的任务的 ID	结果信息	删除 Task
/completion	?{TeamID}	get	TeamID - 小组 ID	小组项目进度信息	查看小组完成情况
	/	post	UserID - 用户 ID, 用于 RBAC UserPassword - 用户密码, 用于 RBAC MilestoneID - 里程碑 ID FileID - 文件 ID	结果信息	新建小组完成情况
	/	put	UserID - 用户 ID, 用于 RBAC UserPassword - 用户密码, 用于 RBAC CompletionID - 完成情况 ID (原) ProjectCompletionInfo - 新增完成情况	结果信息	更改小组完成情况
	/	delete	UserID - 用户 ID, 用于 RBAC UserPassword - 用户密码, 用于 RBAC CompletionID - 完成情况 ID	结果信息	删除小组完成情况
/communication	?{TeamID}	get	TeamID - 小组 ID	交流信息列表	根据小组查看交流信息
	/	post	UserID - 账号 UserPassword - 密码 TeamID - 小组 ID Context - 交流内容 FileID - 文件 ID (nullable)	结果信息	发出交流信息

/studentAccess	?{StudentID}	get	StudentUserID - 用户的 UserID	个人评价	根据 StudentID 获取 Score (学生 id 还是学号?)
	/	post	UserID - 用户 ID, 用于 RBAC UserPassword - 用户密码, 用于 RBAC assess - 评价内容 score - 评价分数 studentInfo - json 格式的 pojo.Student	结果信息	创建新的 StudentAssess
/groupAssess	?{TeamID}	get	teamID - 队伍 ID	队伍评价	根据队伍 ID, 查询队伍成绩
	/	post	UserID - 用户 ID, 用于 RBAC UserPassword - 用户密码, 用于 RBAC assess - 评价内容 teamID - 小组 ID score - 小组成绩	结果信息	创建新的项目评价
/fileAssess	?{FileID}	get	fileID - 查询的文件的 ID	文件评价	查看单个文件的评价
	/	post	UserID - 用户名, 用于 RBAC UserPassword - 用户密码, 用于 RBAC assess - 评价, 用于更新评价 fileInfo - 文件信息, 用于确定更新哪个文件的评价	结果信息	更新文件评价
/project	?{UserID}	get	UserID	项目信息	根据 UserID 查询用户所属项目
	/	post	UserID - 用户 ID, 用于 RBAC UserPassword - 用户密码, 用于 RBAC ProjectName - 项目名字 Description - 项目描述	结果信息	创建新项目
/projects	/	get	无	项目列表	查询所有项目, 只有老师可以
/projectAssig	?{ProjectID}	get	projectID	项目分配信息	根据项目 ID 查询所有选择这个项目的组信息

/projectAssignmen	/	post	UserID - 用户 ID, 用于 RBAC UserPassword - 用户密码, 用于 RBAC ProjectID - 项目 ID TeamID - 小组 ID DeadLine - 项目的 DeadLine	结果信息	小组认领项目
-------------------	---	------	---	------	--------

## 第五部分 界面设计