

卷	号	
卷 内 编 号		
密	级	

〈项目编号〉
 〈思源课堂〉

软件需求规约

〈版本号〉

项 目 承 担 部 门： 西安交通大学

撰 写 人（签名）： 李朝龙

完 成 日 期： 2020/06/19

本文档 使 用 部 门： ☒主管领导 ☒项目组
☐客户（市场） ☒维护人员 ☐用户

评审负责人（签名）： 田丰瑞

评 审 日 期： 2020/06/20

分 类：
 〈模板〉
 使用者：
 〈项目组〉
 文档编号：

HD-RD-306

四川华迪信息技术
 有限公司

Hwadee 华迪

文档信息

标题：软件需求规约
作者：李朝龙
创建日期：2020/06/19
上次更新日期：
版本：1.0
部门名称：

修订文档历史记录

[illegible]

目录

1.	简介	4
1.1	目的	4
1.2	范围	4
1.3	定义、首字母缩写词和缩略语	4
1.4	参考资料	4
1.5	概述	错误！未定义书签。
2.	整体说明	4
3.	具体需求	5
3.1	功能	5
3.1.1	〈功能性需求一〉	错误！未定义书签。
3.2	可用性	7
3.2.1	〈可用性需求一〉	错误！未定义书签。
3.3	可靠性	7
3.3.1	〈可靠性需求一〉	错误！未定义书签。
3.4	性能	7
3.4.1	〈性能需求一〉	错误！未定义书签。
3.5	可支持性	8
3.5.1	〈可支持性需求一〉	错误！未定义书签。
3.6	设计约束	8
3.6.1	〈设计约束一〉	错误！未定义书签。
3.7	联机用户文档和帮助系统需求	8
3.8	购买的构件	错误！未定义书签。
3.9	接口	8
3.9.1	用户界面	错误！未定义书签。
3.9.2	硬件接口	错误！未定义书签。
3.9.3	软件接口	错误！未定义书签。
3.9.4	通信接口	错误！未定义书签。
3.10	许可需求	错误！未定义书签。
3.11	法律、版权及其他声明	40
3.12	适用的标准	错误！未定义书签。
4.	支持信息	错误！未定义书签。

软件需求规约

1. 简介

1.1 目的

《思源课堂软件需求规约》主要目的为收集与组织所有与“思源课堂”相关的需求。我们希望通过《思源课堂软件需求规约》来说明“思源课堂”的各项特性，其中包括几个来自系统用例模型中的用例，以此来说明此特性的功能性需求，以及补充规约中一系列相关的详细需求。

1.2 范围

本文档主要涉及“思源课堂”的要求和设计约束，以及通信组件和检查组件的开发内容和验证方式。

1.3 定义、首字母缩写词和缩略语

SRS: Software Requirements Specification（软件需求规约，以下简称 SRS）

1.4 参考资料

书名/文档	作者	出版社	出版日期
软件工程	王庆育（编著）	清华大学出版社	2004 年 7 月
软件工程	任胜兵、邢琳编	北京邮电出版社	2004 年 5 月
软件工程	张海潘（编著）	人民邮电出版社	2006 年 1 月
UML 基础与 Rose 建模实用教程	谢星星、沈懿卓	清华大学出版社	2008 年 10 月
UML 基础与 Rose 建模教程	蔡敏、徐慧慧、黄炳强编著	人民邮电出版社	2006 年 1 月

2. 整体说明

项目背景：

西安交通大学的课程资源通知与作业处理平台在界面设计方面、人机交互方面、功能性能方面和易用性方面存在诸多问题，对于学校的绝大部分学生（部分老师）而言使用较为不便，他么希望无论是在人机交互或者界面美观度、上传下载资料/作业、查询课程通知/课程成绩等等，都是相对方便且使用起来舒适度极佳的。故我们希望搭建一个面向交大学生、老师的课程通知/作业/成绩管理平台，解决上述的问题，方便学校处理相关事务，最终可以被更多用户接纳认可。

项目目的：

实现一个可以服务于学生、老师、教务员的“课程管理平台”。不同的角色（学生、老师、教务

员)可以有不同的权限,总体上应包含用户的个人信息、课程信息、作业信息、公告信息、资源信息、讨论区信息、成绩信息等等,同时在界面美观度、性能、易用性、人机交互度等方面有较好的改进完善,最终做到可以被绝大多数的学生和老师认可与接受。

项目主要工作:

前期:

①进行校内调研,做需求分析等工作,将不同年级、不同专业的学生与老师对原有“课程管理平台”的使用感受、建议反馈等整理归纳,为后期的改进方向做准备;

②开发项目的主体框架,划分功能块,确定项目范围和产品的开发计划,原型设计。

中期:

全力开发小程序,本地编写各个模块,分模块、功能测试,实现一个产品原型。

后期:

集成源代码,产品上线试运营,对用户反馈做分析并进一步改进产品。

假设:

①项目进行期间不存在人员的退出或更换;

②项目进行期间假设所有人都能按原计划正常工作,不存在请长假的情况;

③假设成本估计没有太大问题,后期不会因为资金原因引起项目崩盘;

④假设人员之间沟通良好,不存在争吵/矛盾;或者发生此情况时,问题可以及时被清理解决;

⑤假设每次开会/集训/工作时,都可以有 5-6 人在场。

约束:

①时间上,必须做到 2020/07/09 前将项目交付完毕;

②成本上,所用的资金必须在前期的预算范围内;

③质量上,必须满足项目前期提出的所有项目需求,必须能够做到让较大部分用户认可我们的产品;

④技术上,团队成员技术方向分散,需要大家在开发中学习多种技术和能力且身兼多职;

实施依据:

本项目是否能够成功实施,只要取决于以下条件:

①团队成员积极配合,为了项目顺利进行,对个人时间合理规划的同时还要为团队做出合理的牺牲,互相配合完成任务。

②必要的工作环境和小程序的运行测试环境都有助于工作的展开。

③团队成员掌握先进的技术用于较快、较好的实现小程序,是该项目成功的保证。

④项目经理管理技术的成熟与否是项目顺利推进的保证。

3. 具体需求

3.1 功能

①课程管理平台界面设计:

每一项功能，包括登录、注册、找回密码、个人信息、课程信息、作业信息、公告信息、资源信息、讨论区信息、成绩信息等等，其界面都应保证干净整洁、易用方便、交互舒适、具备一定的可扩展性等等，给用户良好的使用体验；

②登录功能：

用户通过输入用户名和密码实现登录，同时用户可以选择“记住密码”来实现密码保存，以方便下一次的登录；当用户的用户名和密码一致正确时，跳转到相应的 Dashboard 界面，否则提示密码输入错误；

③注册功能：

用户通过输入用户名、密码、邮箱地址、昵称来实现注册；如果用户名已被注册过，系统提示“该账号已存在”，如果邮箱地址已经被注册过，则提示“该邮箱地址已被注册”；当一切输入无误后，注册成功，系统自动跳回登陆界面；

④密码找回功能：

用户通过输入对应的邮箱地址来实现密码的找回；

⑤Dashboard 界面功能：

- 1) 侧边栏：用于显示用户个人信息和课程信息；
- 2) Upcoming Issue：用于显示用户已经选择的课程的近期作业安排信息；
- 3) Recent Info：用于显示用户所选课程发布的的近期公告、资源、讨论区新内容信息；
- 4) Footer：待定
- 5) 顶级 banner，从左至右需要有：侧边栏扩展按钮，提醒按钮，网页信息按钮，语言按钮、个人信息按钮、聊天按钮等等；（个人信息按钮点击后会出现 logout 选项）

⑥Upcoming Issue 作业/考试通知信息功能：

需要及时将相关信息（作业/问卷/考试等）进行说明，需要能够显示事件的 ID、事件的名称、截止日期、事件的截止内容以及事件当前状态等等；用户如果点击事件的 ID，可以跳转到对应的事件详情说明界面；

⑦Recent Info 功能：

展示教师发布的新的公告、视频、音频、文档资源；要具备新消息的 ID、消息所属内容、消息 Title 和消息所属的时间；

⑧课程信息功能：

需要能够显示课程的公告、资源、讨论板、考试信息和资源等等；课程公告可以进行刷新、点击进入详细公告界面；课程作业可以查看当前课程所发布的各种作业情况信息，对某次作业点击还可以查看作业的详细情况，允许上传、下载作业附件信息，需要有作业截止日期信息并做到根据截止日期和实际时间来对学生进行作业已提交/作业未提交但马上截止/作业提交日期已过等信息，允许查询作业的提交信息数据汇总信息，允许书写作业草稿，允许发送提交作业内容（或上传作业文件）；讨论板可以汇总参与这门课程的所有

同学所提交的讨论信息，讨论信息要显示出发布信息者、发布时间、回复对象等等；资源方面，老师可以上传相关资源文件（课件、视频）等，学生可以进行下载观看。

3.2 可用性

①易操作，易理解

尽量从用户角度出发，以方便使用本产品。如：在第一次使用某些功能时会有提示框指导用户如何操作。用户最近的活动会优先显示在主页上，用户上一次的发起的活动的的数据选择会保留并默认延用至下一次活动。

②稳定性

系统会更新增加功能，不断修复 bug，使系统更加稳定和多样化。

③容错能力

系统具有一定的容错和抗干扰能力，在非硬件故障或非通讯故障时，系统能够保证正常运行，并有足够的提示信息帮助用户有效正确地完成任务。在弱网络下也能实现基础功能。

④并发性

能够保证一定量的用户同时在线使用。后期盈利足够时考虑添设服务器以增大可同时在线的用户量。

⑤操作完成时有统一规范的提示信息。

例如确定活动创建时，系统弹出提示框，用户可查看自己已经选择了的信息。用户点击确认后，系统才创建活动。用户在支付时，会有支付确认以及身份验证。每日签到可以进行提示。

⑥控制必填项

本系统需要用户进行微信授权和身份验证，以保证用户信息完整和支付安全。

⑦用户可自定义

为了满足业务的不断变化，一些重要的参数应该可以灵活设置，非重要参数可在下一次大的更新中更改。

⑧帮助和操作指南

一般会给首次使用的用户在小程序中予以指导。

3.3 安全性

①重要数据加密

对一些重要的数据利用可靠的加密技术进行加密，例如用户的密码以及用户的个人信息等。

②记录日志

本系统应该能够记录系统运行是所发生的所有错误，包括本机错误和网络错误。错误记录便于查找错误的原因。日志同时记录用户的关键性操作信息。

③ 故障处理

正常使用时不应出错，若运行时出现不可恢复的系统错误，也必须保证数据库完整。且出现的故障必须记录于日志中。

④ 权限控制

根据不同用户角色，设置相应权限，用户的重要操作都做成相应的日志记录以备查看，没有权限的用户禁止使用系统。用户只能管理自己的个人信息，管理员可以管理用户和自己的个人信息。

⑤ 数据备份

允许用户进行数据的备份和恢复，以弥补数据的破坏和丢失。

3.4 性能

阐述不同的应用领域对产品性能的需求，并解释它们的原理以帮助开发人员做出合理的设计选择，确定相互合作的用户数或者所支持的操作，响应时间以及与实时系统的时间关系；还要定义容量需求，例如存、储器和磁盘空间的需求或者存储在数据库中表的最大行数。也可能需要针对每个功能需求或特性分别陈述其性能需求：

①用户一般响应时间不超过 1 秒，上传作业或者图片等不超过 5 秒；

②如果页面长时间无响应，应提示加载错误后让用户刷新平台界面；

③按照规定的格式输入，否则系统提示错误并要求重新输入；

④要求数据库有很好的更新能力，能够适应迭代开发；

⑤平台不宜太复杂，方便用户加载使用；

- ⑥弱网络下能顺利加载基础功能;
- ⑦能够承载较大数量的用户并发使用度;
- ⑧易于扩展、迭代、修改完善;
- ⑨可以达到较高的吞吐量。

3.5 可维护性需求

- ①保留系统对应的版本的源代码。
- ②代码一定要有注释，特别是条件选择部分，一定备注是什么功能。
- ③清晰的系统结构和命名规范，界面规范，提示和帮助信息规范，友好的错误提示信息，可以帮助用户自己找原因，自己维护系统。

3.6 设计约束

- ①开发周期短：
三个星期的开发时间需要开发者合理规划时间，做到多项任务并发。
- ②人员技术和数量有限：
团队成员的技术水平不够成熟且只有 6 人，需要在开发中学习多种技术和能力且身兼多职。
- ③代码规范约束：
项目团队成员需要采用一定的代码规范约束，以保证后期代码的合并顺利。

3.7 接口

声明

由于后端数据库变量命名采用以小写字母组成的英文单词和下划线分割构成表项名称，故所有前后端发送的内容建议以小写英文单词加下划线命名消息。

3.7.1 Login 接口

接口路径：

```
/Login
```

Js 函数路径：

```
js/myjs.js?v=1
```

请求方法：

```
Post
```

正向接口格式：

```
data:JSON.stringify({  
    "status": "0",
```



```
        "message": "login",

        "user_name": user_name,

        "pass_word": pass_word,

    }),
```

Username, Password, 分别表示用户名, 密码。

反向接口格式:

```
{

    "success_value": 0, // 1 表示成功, 0 表示失败

    "role": "teacher/student/manager", //分别表示教师用户, 学生用户, 管理员

    "role_id": person_id, //教师, 学生或管理员 id

    "fail_content": "错误详细信息, 由后端返回", //错误详细信息: 邮箱不存在

}
```

数据查询格式:

```
SET @login_username = 'tianfengrui';

SET @login_password = 'tianfengrui';

SELECT

    student_password AS correct_password

FROM

    students
```

WHERE

students.student_username = @login_username

UNION

SELECT

teacher_password

FROM

teachers

WHERE

teachers.teacher_username = @login_username;

3.7.2 Register 接口

注意：本接口在 1.0 版本后正向接口格式已更改。

该接口出现在 [3.1.2 节注册界面](#)。

接口路径：

/Register

接口 js 函数路径：

Js/myjs.js?v=1

请求方法：

Post

正向接口格式：

```
data:JSON.stringify({
  "status": "0",
  "message": "register",
  "user_name": user_name,
```

```
"pass_word": pass_word,  
  
"email": email,  
  
"full_name": full_name,  
  
"role": "teacher/student/manager" //分别指明是教师，学生，管理员角色  
  
}),
```

反向接口格式：

```
{  
  
  "success_value": 0, // 1 表示成功，0 表示失败  
  
  "fail_content": "错误详细信息，由后端返回", //错误详细信息：邮箱不存在  
  
}
```

Sql 语句格式：

第一步：判断用户名是否重复：

```
/* 6.3 接口 */  
  
/* 第一步：判断用户名是否重复 */  
  
USE SummerProject;  
  
SET @new_username = "tianfengrui";  
  
SELECT  
  
  student_id AS user_id,  
  
  student_username AS username  
  
FROM  
  
  students
```

WHERE

students.student_username = @new_username

UNION

SELECT

teacher_id AS user_id,

teacher_username AS username

FROM

teachers

WHERE

teachers.teacher_username = @new_username;

若无重复，插入数据：

教师用户：

/* 第二步：教师用户插入 */

INSERT INTO SummerProject.teachers (

teacher_id,

teacher_name,

teacher_username,

teacher_password,

teacher_email,

insert_date)

VALUES

(

"tch20000110", "caodahua" , "dddd", "dddd" , "3434@223.com", LOCALTIME()

);

学生用户：

```
/* 第二步：学生用户插入 */  
  
INSERT INTO SummerProject.students (  
  
    students.student_id,  
  
    student_name,  
  
    student_username,  
  
    student_password,  
  
    student_email,  
  
    insert_date  
  
)  
  
VALUES (  
  
    "stu20000110",  
  
    "caod3ahua",  
  
    "ddd3d",  
  
    "ddd3d",  
  
    "34334@223.com",  
  
    LOCALTIME()  
  
);
```

3.7.3 PasswordReminder 接口

该接口出现在 [3.1.3 节密码找回界面](#)。

接口路径：

```
/PasswordReminder
```

接口 js 函数路径：

```
Js/myjs.js?v=1
```

请求方法：

Post

正向接口格式:

```
data: JSON.stringify({  
  "status": "0",  
  "message": "password_reminder",  
  "email": email,  
}),
```

反向接口格式:

```
{  
  "success_value": 0, // 1 表示成功, 0 表示失败  
  "fail_content": "错误详细信息, 由后端返回", //错误详细信息: 邮箱不存在  
}
```

Sql 语句格式:

```
/* 6.4 PasswordReminder */  
  
USE SummerProject;  
  
SELECT  
  teachers.teacher_username AS username,  
  teachers.teacher_password AS PASSWORD,  
  teachers.teacher_email AS email  
  
FROM  
  teachers  
  
WHERE
```

```
teacher_email = "648251476@qq.com"

UNION

SELECT

    student_username AS username,

    student_password AS PASSWORD,

    students.student_email AS email

FROM

    students

WHERE

    student_email = "648251476@qq.com";
```

3.7.4 DashboardMenu 接口

该接口出现在 3.2.1.3 节 [Courses 栏下内容](#)

接口路径:

```
/DashboardMenu
```

接口 js 函数路径

```
Js/myjs.js?v=1
```

请求方法:

```
Post
```

正向接口格式:

```
data: JSON.stringify({

    "status": "0",

    "user_id": user_id,

    "message": "dashboard_menu",
```

```
}},
```

反向接口格式:

```
{
  "avatar":avatar_path,
  "user_name": user_name,
  "id": "courses",
  "value": [
    {
      "id": "course1",
      "course_name": course_name,
      "course_id": course_id
    },
    {
      "id": "course2",
      "course_name": course_name,
      "course_id": course_id
    },
    {
      "id": "course3",
      "course_name": course_name,
      "course_id": course_id
    }
  ],//当前学期课程数量不定，需要用循环遍历，id 自增
}
```



```
}
```

注意：id 为 courses 项的 value 项为列表，其项目数量不确定。

Course_name 项存储课程名称，显示到侧边栏上，Course_id 项存储访问该课程的网址

Sql 语句格式：

```
/* 6.5 DashboardMenu */  
  
USE SummerProject;  
  
SELECT  
  
    courses.course_id,  
  
    courses.course_name,  
  
    courses.course_hierarchy,  
  
    courses.course_semester  
  
FROM  
  
    courses  
  
WHERE  
  
    courses.course_id IN (  
  
        SELECT  
  
            students_courses.course_id  
  
        FROM  
  
            students_courses  
  
        WHERE  
  
            student_id = "stu20200202"  
  
    )  
  
AND courses.course_hierarchy = "2019"  
  
AND courses.course_semester = 'spring';
```

3.7.5 UpComingIssue 接口

该接口出现在 [3.2.2 节 Upcoming Issue 详细设计](#)。

接口路径:

```
/UpcomingIssue
```

接口 js 函数路径

```
Js/myjs.js?v=1
```

正向接口格式:

```
data: JSON.stringify({
    "status": "0",
    "message": "upcoming_issue",
}),
```

反向接口格式

```
{
    "id": "issues",
    "total_num": num_of_total_issues,
    "value": [
        {
            "id": "issue_id",
            "value": issue_id,
            "issue_id_path": issue_id_path,
            "course": course,
            "deadline": deadline,
            "content": issue_title,
            "status": status //expired, on_schedule, submitted, On_schedule_yellow
```

```

    },
    {
        "id": "issue_id",
        "value": issue_id,
        "issueidpath": issue_id_path,
        "course": course,
        "deadline": deadline,
        "content": issue_title,
        "status": status //expired, on_shedule, submitted, on_shedule_yellow

    },
    //项目数量不定，需要遍历访问
]
}

```

3.7.6 RecentInfo 接口

该接口出现在 [3.2.3 节 Recent Info 详细设计](#)。

接口路径：

```
/RecentInfo
```

接口 js 函数路径

```
Js/myjs.js?v=1
```

正向接口格式：

```

data: JSON.stringify({
    "status": "0",
    "message": "recent_info",

```

```
}},
```

反向接口格式

```
{  
  "id": "recent_info",  
  "value": [  
    {  
      "id": "info_id1",  
      "value":info_id,  
      "info_id_path": info_id_path,  
      "course":course,  
      "datetime":datetime,  
      "content":info_title,  
    },  
    {  
      "id": "info_id2",  
      "value":info_id,  
      "issue_id_path": issue_id_path,  
      "course":course,  
      "deadline":deadline,  
      "content":info_title,  
    },  
  ],  
}
```

//项目数量不定，需要遍历访问

```
]

}
```

3.7.7 GetCoursePanel 接口

该接口出现在 [3.4.1.2 节 Folder 栏下部分](#)。

接口路径:

```
/GetCoursePanel
```

接口 js 函数路径

```
Js/myjs.js?v=1
```

正向接口格式:

```
data:JSON.stringify({
    "status": "0",
    "message": "get_course_panel",
    "course_id": course_id,
}),
```

反向接口格式

```
{
    "id": "course_panel",
    "value": [
        {
            "panel_id": panel_id,
            "panel_name": panel_name,
```

```
    },  
    {  
        "panel_id": panel_id,  
        "panel_name": panel_name,  
    },  
  
    //项目数量不定，需要遍历访问  
  
]  
}
```

3.7.8 GetCourseAnnouncements 接口

该接口出现在 [3.4.2.1 节公告详情页设计](#)。

接口路径：

```
/GetCourseAnnouncements
```

接口 js 函数路径

```
Js/myjs.js?v=1
```

正向接口格式：

```
data: JSON.stringify({  
    "status": "0",  
    "message": "get_course_announcements",  
    "course_id": course_id,  
}),
```

反向接口格式

```
{
  "id": "course_announcements",
  "total_num": num_of_total_announcements,
  "value": [
    {
      "announcement_title": announcement_title, //公告标题
      "announcement_id": announcement_id, //公告访问具体路径，用户点击公告标题后向后端接口发送该路径进入公告具体信息页
      "announcement_content": announcement_content, //公告内容
    },
    {
      "announcement_title": announcement_title,
      "announcement_id": announcement_id,
      "announcement_content": announcement_content,
    },
    //项目数量不定，需要遍历访问
  ]
}
```

3.7.9 GetAnnouncement 接口

该接口出现在 [3.4.2.1.2 节课程公告页设计](#)。

接口路径：

```
/GetAnnouncement
```

接口 js 函数路径

```
Js/myjs.js?v=1
```

正向接口格式:

```
data: JSON.stringify({
    "status": "0",
    "message": "get_announcement",
    "course_id": course_id,
    "announcement_id": announcement_id //注意: 这个变量在 get_course_announcement 中出现过。
}),
```

反向接口格式

```
{
    "announcement_title": announcement_title, //公告标题
    "author": teacher_name, //公告发布教师名称
    "author_email": teacher_email, //公告发布教师邮箱
    "author_avatar": authon_avatar, //默认为初始头像
    "announcement_content": announcement_details, //这里有点争议, 我设想返回的是 html 格式文档
    "datetime": annoucement_datetime, //教师发布该公告的时间
}
```

3.7.10 GetCourseHomeworks 接口

该接口出现在 [3.4.2.2.1 作业展示具体设计节](#)。

接口路径:

```
/GetCourseHomeworks
```


接口 js 函数路径

```
Js/myjs.js?v=1
```

正向接口格式:

```
data: JSON.stringify({
    "status": "0",
    "message": "get_course_homeworks",
    "course_id": course_id,
}),
```

反向接口格式

```
{
    "id": "course_homeworks",
    "total_num": num_of_total_homeworks,
    "value": [
        {
            "homework_title": homework_title, //作业标题
            "homework_id": homework_id, //作业访问具体路径, 用户点击作业标题后向后端接口发送该路径
            进入公告具体信息页
            "homework_content": homework_content, //作业内容
            "homework_deadline": homework_deadline, //作业截止日期
        },
        {
            "homework_title": homework_title, //作业标题
            "homework_id": homework_id, //作业访问具体路径, 用户点击作业标题后向后端接口发送该路径
            进入公告具体信息页
            "homework_content": homework_content, //作业内容
            "homework_deadline": homework_deadline, //作业截止日期
        },
    ],
}
```

```
        //项目数量不定，需要遍历访问

    ]
}
```

3.7.11 GetHomework 接口

该接口出现在 [3.4.2.2 课程作业界面具体设计](#) 节中。

接口路径：

```
/GetHomework
```

接口 js 函数路径

```
Js/myjs.js?v=1
```

正向接口格式：

```
data: JSON.stringify({
    "status": "0",
    "message": "get_homework",
    "course_id": course_id,
    "homework_id": homework_id, //注意这个变量在 GetCourseHomeworks 中出现过
}),
```

反向接口格式

```
{
    "homework_title": homework_title, //作业标题
    "author": teacher_name, //作业发布教师名称
    "author_email": teacher_email, //作业发布教师邮箱
```

```

    "author_avatar": authon_avatar, //默认为初始头像

    "homework_deadline": datetime //作业的ddl

    "homework_content": announcement_details, //这里有点争议，我设想返回的是html 格式文档

    "homework_attachment": [

        {

            "attachment_path": path_to_attachment_in_webserver, //用户通过点击附件图标跳转到该路径实现附件下载

            "attachment_name": attachment_name,

        },

        {

            "attachment_path": path_to_attachment_in_webserver,

            "attachment_name": attachment_name,

        },

        {

            "attachment_path": path_to_attachment_in_webserver,

            "attachment_name": attachment_name,

        },

        //项数不定需要遍历，如果没有附件，列表为空。

    ]

    "datetime": annoucement_datetime, //教师发布该公告的时间

}

```

3.7.12 SubmitHomework 接口

该接口出现在 [3.4.2.2.3 课程作业提交界面具体设计](#) 节中。

接口路径：

```
/SubmitHomework
```

接口 js 函数路径

Js/myjs.js?v=1

正向接口格式:

```
data: JSON.stringify({
  "status": "0",
  "message": "submit_homework",
  "course_id": course_id,
  "homework_id": homework_id, //注意这个变量在 GetCourseHomeworks 中出现过
  "is_draft": "yes" //yes 表示草稿, no 表示提交作业
  "homework_content": homework_content, //用户在网页上写的内容保存到这里, html 格式
  "have_attachments": "yes" //yes 表示有上传附件, no 表示没有上传附件
  "homework_attachments": [
    {
      "file_name": file_name,
      "file_path": file_path, //在 FileUpload 接口获取的文件路径
    },
    {
      "file_name": file_name,
      "file_path": file_path, //在 FileUpload 接口获取的文件路径
    },
    {
      "file_name": file_name,
      "file_path": file_path, //在 FileUpload 接口获取的文件路径
    },
    //文件数量不固定
  ]
})
```

```
}},
```

反向接口格式

```
{  
    "success_value": 0, //1 表示成功, 0 表示失败  
    "datetime": datetime, //作业成功提交的具体时间  
    "fail_content": "错误详细信息, 由后端返回", //错误详细信息: 服务器连接失败, 删除失败, 服务器错误.....  
}
```

3.7.13 GetExistedHomework 接口

该接口出现在 [3.4.2.2.3 课程作业提交界面具体设计](#) 节。

接口路径:

```
/GetExistedHomework
```

接口 js 函数路径:

```
Js/myjs.js?v=1
```

正向接口设计

```
data: JSON.stringify({  
    "status": "0",  
    "message": "get_existed_homework",  
    "course_id": course_id,  
    "homework_id": homework_id,  
}),
```

反向接口设计

```

{
    "homework_status": "drafted, submitted, null, expired"//drafted 表示已保存草稿, submitted
    d 表示已提交, null 表示未提交无草稿, expired 表示作业超过截至时间

    "homework_content":homework_content,//用户在网页上写的内容保存到这里, html 格式, 若状态为 null
    或状态 expired 并且未提交作业此项为空,

    "submitted_time": datetime, //作业草稿或提交作业的具体时间

    "homework_deadline": datetime, //该作业的 ddl

    "have_attachments": "yes" //yes 表示有上传附件, no 表示没有上传附件

    "homework_attachments":[
        {
            "file_name": file_name,

            "file_path": file_path, //在 FileUpload 接口获取的文件路径
        },
        {
            "file_name": file_name,

            "file_path": file_path, //在 FileUpload 接口获取的文件路径
        },
        {
            "file_name": file_name,

            "file_path": file_path, //在 FileUpload 接口获取的文件路径
        },
        //文件数量不固定, 若状态为 null 或状态 expired 并且未提交作业此项为空
    ]
}

```

本项接口有争议：如何实现浏览器实时显示上传进度？

3.7.14 FileUpload 接口

该接口出现在 [3.4.2.2.3 课程作业提交界面具体设计](#) 节中。

接口路径:

```
/FileUpload
```

接口 js 函数路径:

```
Js/myjs.js?v=1
```

限制: 若是作业附件上传, 单个文件大小不超过 1024MB。

正向接口格式:

```
data: JSON.stringify({
    "status": "0",
    "message": "file_upload",
    "role": "student\teacher", //学生在 homework 界面上传文件则为学生角色, 若老师上传文件为老师角色
    "course_id": course_id, //学生或教师上传文件时所在的课程的课程名
    "related_id": related_id, //若为学生角色, 则该项存储 homework_id, 若为教师角色, 该项存储 XXXX_id
    "file_name": file_name,
    "file_content": file_content, //不确定是否应该这么写
}),
```

反向接口格式:

```
{
    "success_value": 0, //1 表示成功, 0 表示失败
    "fail_content": "错误详细信息, 由后端返回", //错误详细信息: 服务器连接失败, 上传失败, 服务器错误, 文件过大, 文件类型不支持.....
    "file_path": file_path, //若上传成功, 该项返回文件在服务器保存的路径, 若失败则为 null
}
```

3.7.15 FileDelete 接口

该接口出现在 [3.4.2.2.3 课程作业提交界面具体设计](#) 节中。

接口路径:

```
/FileDelete
```

接口 js 函数路径:

```
Js/myjs.js?v=1
```

正向接口格式:

```
data: JSON.stringify({
  "status": "0",
  "message": "file_delete",
  "role": "student\teacher", //学生在 homework 界面上传文件则为学生角色，若老师上传文件为老师角色
  "course_id": course_id, //学生或教师上传文件时所在的课程的课程名
  "related_id": related_id, //若为学生角色，则该项存储 homework_id，若为教师角色，该项存储 XXXX_id
  "file_name": file_name,
  "file_path": file_path, //上传文件成功时返回的文件路径
}),
```

反向接口格式

```
{
  "success_value": 0, //1 表示成功，0 表示失败
  "fail_content": "错误详细信息，由后端返回", //错误详细信息：服务器连接失败，删除失败，服务器错误.....
}
```

3.7.16 FileDownload 接口

本项接口存在争议，如何实现下载进度条？（已解决）

该接口出现在 [3.4.2.5 资源详情页](#) 节中。

接口路径:

```
/FileDownload
```

接口 js 函数路径:

```
Js/myjs.js?v=1
```

正向接口格式:

```
data: JSON.stringify({
    "status": "0",
    "message": "file_download",
    "file_name": file_name,
    "file_path" file_path, //上传文件成功时返回的文件路径
}),
```

反向接口格式

```
{
    "file_content":file_content //文件内容
}
```

3.7.17 FilesDownload 接口

本项接口存在争议，如何实现下载进度条？

该接口出现在 [3.4.2.5 资源详情页](#) 节中。

接口路径:

```
/FilesDownload
```

接口 js 函数路径:

Js/myjs.js?v=1

正向接口格式:

```
data: JSON.stringify({
  "status": "0",
  "message": "files_download",
  "files":[
    {
      "file_name":file_name,
      "file_path":file_path,
    },
    {
      "file_name":file_name,
      "file_path":file_path,
    },
    {
      "file_name":file_name,
      "file_path":file_path,
    },
  ]
}),
```

反向接口格式

```
{
  "files":[
```

```
{
  "file_content":file_content,
},
{
  "file_content":file_content,
},
{
  "file_content":file_content,
},
]
```

3.7.18 GetCourseResources 接口

该接口出现在 [3.4.2.5 资源详情页](#) 节中。

接口路径:

```
/GetCourseResources
```

接口 js 函数路径:

```
Js/myjs.js?v=1
```

正向接口格式:

```
data: JSON.stringify({
  "status": "0",
  "message": "get_course_resources",
  "course_id": course_id,
}),
```

反向接口格式

```
{
  "id":course_resources,
  "total_num": num_of_total_resources,
  "value":[
    {
      "resource_name": resource_name,
      "resource_path": path_to_resource_on_server,
      "resource_size": size_of_resource,
      "resource_upload_time": upload_time_of_resource,
      "resource_type": type_of_resource, // mp4, mp3, rar, pdf, doc, ppt, etc....
    },
    {
      "resource_name": resource_name,
      "resource_path": path_to_resource_on_server,
      "resource_size": size_of_resource,
      "resource_upload_time": upload_time_of_resource,
      "resource_type": type_of_resource, // mp4, mp3, rar, pdf, doc, ppt, etc....
    },
    {
      "resource_name": resource_name,
      "resource_path": path_to_resource_on_server,
      "resource_size": size_of_resource,
      "resource_upload_time": upload_time_of_resource,
      "resource_type": type_of_resource, // mp4, mp3, rar, pdf, doc, ppt, etc....
    },
  ]
}
```

```
}
```

3.7.19 NewCourseIssue 接口

该接口出现在 [4.4.2.1.1 新建公告界面](#) 节中。

接口路径:

```
/NewCourseIssue
```

接口 js 函数路径:

```
Js/myjs.js?v=1
```

正向接口格式:

```
data:JSON.stringify({
    "status": "0",
    "message": "new_course_issue",
    "type": "announcement\homework\examination\discussion\resource",//发布的是作业、公告、考试、
    讨论区、资源
    "title": "issue_title",
    "content": "content",
    "attachment":[
        {
            "file_name": file_name,
            "file_path": file_path,
        },
        {
            "file_name": file_name,
            "file_path": file_path,
        },
        {
            "file_name": file_name,
```

```
        "file_path": file_path,

    },

]

}),
```

反向接口格式:

```
{

    "success_value": 0, //1 表示成功, 0 表示失败

    "fail_content": "错误详细信息, 由后端返回", //错误详细信息: 服务器连接失败, 删除失败, 服务器错误.....

}
```

3.7.20 GetPrevCourses 接口

该接口出现在 [3.3PrevCourses](#) 节中。

接口路径:

```
/GetPrevCourses
```

接口 js 函数路径:

```
Js/myjs.js?v=1
```

正向接口格式:

```
data:JSON.stringify({

    "status": "0",

    "message": "get_prev_courses",

    "role": student/teacher/manager,

    "person_id": person_id,

}),
```

反向接口格式:

```
{  
  "id": "prev_courses",  
  "total_num": num_of_total_courses,  
  "value": [  
    {  
      "course_title": course_title,  
      "course_teacher": course_teacher,  
      "course_hierarchy": course_hierarchy,//2016,2017 级  
      "course_college": course_college,//开课学院  
    },  
    {  
      "course_title": course_title,  
      "course_teacher": course_teacher,  
      "course_hierarchy": course_hierarchy,//2016,2017 级  
      "course_college": course_college,//开课学院  
    },  
    {  
      "course_title": course_title,  
      "course_teacher": course_teacher,  
      "course_hierarchy": course_hierarchy,//2016,2017 级  
      "course_college": course_college,//开课学院  
    },  
  
    //项目数量不定，需要遍历访问  
  ]  
}
```

}

3.8 法律、版权及其他声明

凡未经允许盗用、窃取公司项目机密、侵犯知识产权者，公司有责依法对其进行追究。