

**软件工程课程设计报告**

( 2019 -- 2020 学年 第一学期 )

课程名称： 软件工程课程设计

题 目： 移动教学平台

院 系：控制与计算机工程学院

班 级： 软件1702

组 号：

组 长： 于博淞

组 员： 穆泽睿 赵鑫 向星辰

指导教师： 王素琴

设计周数： 两周

小组成绩：

日期：2019 年 1月 4日

**《软件工程》课程设计**

**任 务 书**

**一、 目的、要求**

通过软件开发的实践训练，进一步掌握软件工程的方法和技术，提高软件开发的实际能力，培养工程设计能力和综合分析、解决问题的能力。

具体如下：

1. 学习和实践在分析和设计计算机应用系统所需要的知识,包括面向对象的系统分析与设计,编码和测试方面的知识；
2. 熟悉自动化的软件开发工具，将其运用于软件开发的全过程；
3. 进一步加强和提高软件工程文档的编写能力；
4. 培养协作能力和团队精神。

**二、 主要内容**

1. 综合运用软件工程方法、技术和工具完成宿舍管理系统的分析与设计任务；
2. 使用EA或StarUML等作为需求分析与设计的建模工具，进行系统的静态建模和动态建模。
3. 利用对象模型自动生成数据模型,自动建立数据库；
4. 选择PHP语言对关键的模块进行界面层的设计并给出实现；
5. 撰写课程设计报告。

**三、任务分配**

|  |  |  |  |
| --- | --- | --- | --- |
| **任务** | **负责人** | **参与人** | **备注** |
| 系统分析-类图 | 于博淞 | 向星辰 |  |
| 系统分析-用例图 | 穆泽睿 | 向星辰 |  |
| 系统分析-顺序图 | 向星辰 | 于博淞 |  |
| 系统设计—体系结构设计、问题域子系统设计 | 赵鑫 | 穆泽睿 |  |
| 系统设计—数据管理子系统 | 赵鑫 | 穆泽睿 |  |
| 界面设计 | 向星辰 | 于博淞 | H5、C3 |
| 编程—作业管理 | 穆泽睿 | 赵鑫、于博淞 | PHP |
| 编程—课程管理 | 穆泽睿 | 赵鑫、于博淞 | PHP |
| 编程—前后端连接 | 于博淞 | 赵鑫、穆泽睿 | PHP、JS |
| 课程设计报告 | 于博淞 | 穆泽睿、赵鑫、向星辰 |  |

**四、 进度计划**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **设计内容名称** | **完成时间** | **备注** |
| 1 | 分组及确定题目 | 1个工作日 |  |
| 2 | 初步的需求分析与设计建模, 确定实现平台，并搭建环境 | 2个工作日 |  |
| 3 | 详细的需求分析与设计建模 | 2个工作日 | 进行中期检查 |
| 4 | 关键模块的实现与测试 | 3个工作日 |  |
| 5 | 编写课程设计报告 | 1个工作日 |  |
| 6 | 验收检查及评定成绩 | 1个工作日 |  |

**五、 设计成果要求**

1. 建立系统分析与设计模型；
2. 初步建立系统原型，实现关键的功能；
3. 编写课程设计报告。

**六、 考核方式**

1. 系统演示及讲解

占50%。

1. 设计报告

占50%。

指导教师：王素琴

日 期： 2018年 12 月 28 日

**《软件工程》课程设计**

**成 绩 评定**

**一、 指导教师评语**

**二、 成绩**

|  |  |  |  |
| --- | --- | --- | --- |
| **学号** | **姓名** | **成绩** | **备注** |
| 120171080702 | 于博淞 |  |  |
| 120171080403 | 穆泽睿 |  |  |
| 120171080423 | 赵鑫 |  |  |
| 120171080729 | 向星辰 |  |  |

指导教师：王素琴

日 期： 2019 年 1 月 4 日

**摘要：**随着电脑的普及与使用，现在的管理也提升了一个档次，渐渐实现了无纸化办公，即从原来的人工记录管理模式转变为电脑一体化管理。高校是科研的阵地，学生作业管理也应该一改传统的人工管理，更加信息化，时代化，节省人力物力，提高效率。基于这一点，开发此移动教学平台。

移动教学平台，是以学生提交作业为实例而设计的一种实用型系统。本系统最大的特点是通用性、简单操作性，适用于计算机专业学生和老师关于作业的提交和审查需求。随着学生数量的增多，老师作业审查的负担越来越重，为了让老师和学生能从繁重的工作中解脱出来，实现无纸化办公；使工作更有条理，更方便，更有效率而开发出这个移动教学平台。

该程序有学生基本信息管理、学生作业提交、作业查重、教师下载学生提交的、作业提交情况统计等功能。本系统依据学生日常作业提交和老师作业批改等需求而设计，  
系统开发过程中我们使用了xampp建站集成软件包（即Apache作Web服务器，MySQL作数据库，PHP写后端程序），并采用了ThinkPHP5.0作为框架进行开发，最后设计出的系统能满足日常作业提交和审查等基本功能。

**关键词：**作业管理，MVC,代码查重

**一、课程设计的目的与要求**

1.1 课程设计的目的

1.学习和实践在分析和设计计算机应用系统所需要的知识,包括面向对象的系统分析与设计,编码和测试方面的知识；

2.进一步加强和提高软件工程文档的编写能力；

3.培养协作能力和团队精神。

1.2 课程设计的要求

1.综合运用软件工程方法、技术和工具完成学生宿舍管理系统的分析与设计任务；

3.对移动教学平台系统中关键的模块进行界面层的设计并给出实现,选择Java平台；

4.撰写课程设计报告。

**二、设计正文**

**1. 概述**

1.1 设计课题

该系统为移动教学平台。

1.2 系统目标

调研学生老师日常提交作业的情况，建立一个与实际较为吻合的作业教学管理系统。

1.3系统开发与运行环境

（1）数据库管理软件：MySqL

（2）编程工具：Eclipse

（3）操作系统：Windows系列

（4）开发语言：PHP

**2. 系统需求分析**

**2.1系统需求概述**

主要从学生用户、教师用户两方面着手：

**教师用户：**

（1）用户注册

（2）用户登录

（3）用户可以创建课程。

（4）用户可以查询、修改个人信息。

（5）用户可以修改课程信息。

（6）用户可以布置作业。

（7）用户可以添加课程成员。

（8）用户可以布置作业。

（9）用户可以查看作业提交情况。

（10）用户可以下载学生用户提交的作业

（11）用户可以查看某一门课程学生提交作业情况的统计表格

（12）用户可以查看学生提交作业的重复率情况

**学生用户：**

（1）用户注册

（2）用户登录

（3）用户可以查询、修改个人信息。

（4）用户可以申请加入、退出课程。

（5）用户可以提交作业

（6）用户可以修改用户自己密码。

**2.2 用例图**

对于整个用例图，外部角色有教师用户和学生用户。教师有创建课程、删除课程、审核申请、创建作业、下载学生作业、查重作业等用例，而学生用户有申请课程、上交作业、下载已上传的作业等用例。

由于总的用例图过大，所以将重点展示本次该系统更新的功能的用例图（图1与图2）。

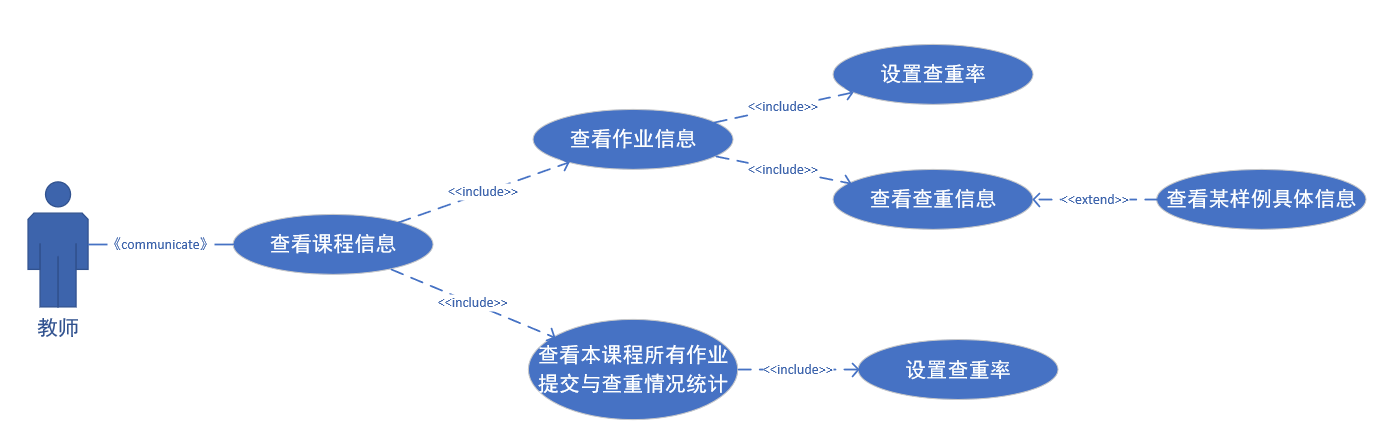


图1 教师查重作业用例图

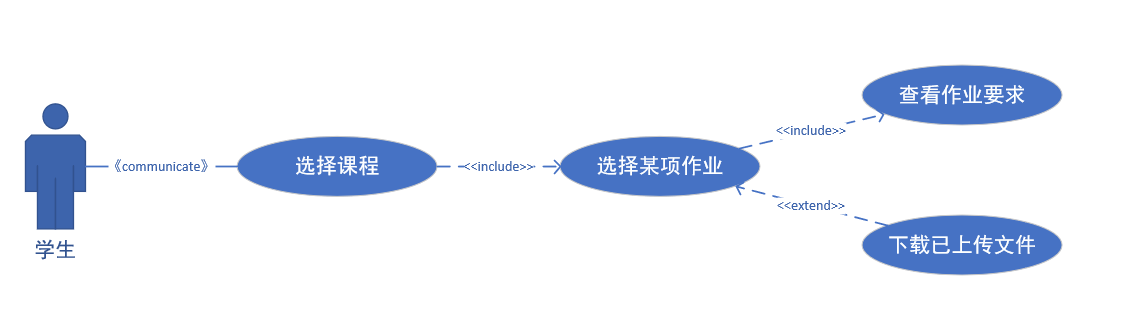


图2 学生下载自己提交的作业用例图

**2.3类图**

经过需求分析，得到移动教学平台的类图。

本系统包含EmptyClass、Checkuser、Bbs、File、Homework、Index、Lesson、Login、MOSS、Myanswer、Myquestion、ResourceModel、Sharefile、ShareFileRecord、Teacher、User、Bbs\_answerModel、Bbs\_questionModel、Homework\_submitModel、HomeworkModel、Lession\_applyModel、Lession\_userModel、LessionModel、ResourceModel、ShareModel、statistic\_dowloadModal、statistic\_previewModal、UserModel等共28个类，分为两种：Controller和Model，其中Model存放有关数据库内容，Controller存放有关后台操作内容。

在此28类中，有关查重和统计的代码集中在以下几个类：

Teacher为教师操作类，包含ajax\_chachong（负责控制查重功能）、ajax\_getHomeworkSubmitList（负责获得学生提交某次作业的情况列表功能）等。

Lesson为课程操作类，主要使用了ajax\_getHomeworkList（负责获得作业列表功能）。

Homework为专作业操作类，主要使用了getCopyNum（获得某次课程某学生所有作业重复次数）。

MOSS为斯坦福大学提供的已封装好的类，其主要功能为返还上传作业中涉嫌抄袭的文件。



图3 查重部分类图

各类中的关系：Teacher继承自Checkuser（负责查看用户为老师还是学生），而Checkuser继承自EmptyClass，所以他们之间为泛化关系。Teacher和Lesson之间是关联关系，他们是多对多的关系；Lesson和Homework之间也是关联关系，他们是一对多的关系，即一个lesson可以有多个homework；Homework和Moss之间也是关联关系，他们是一对一的。

**2.4 顺序图**

教师查重作业的顺序图如图4所示，首先是教师查看单项作业统计的界面，该界面会去数据库找出相应信息，教师可再设置查重率，作业统计界面将该数据传给查重界面，查重界面便可从moss中读取相应数据。

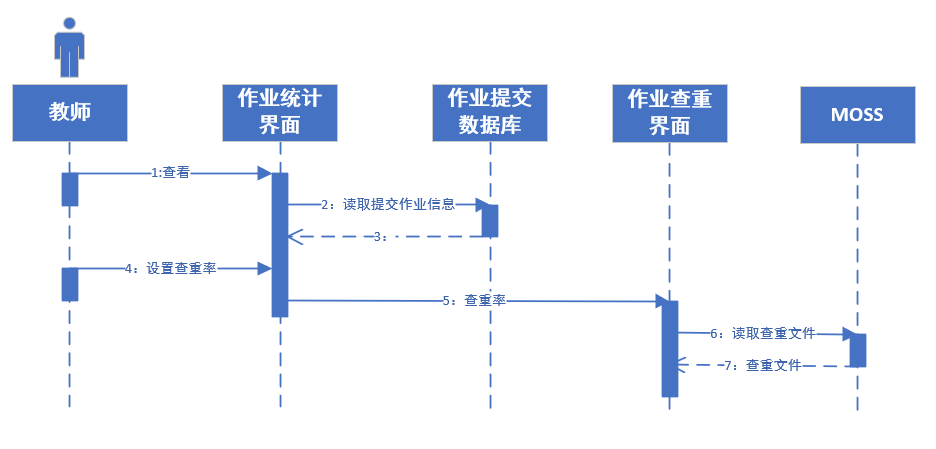


图4 教师查重顺序图

教师查看作业信息统计的顺序图如图5所示，首先是教师查看所有作业统计的界面，该界面会去数据库找出相应信息，教师可再设置查重率，界面会根据查重率在数据库中查看信息并返回到界面给教师查看。

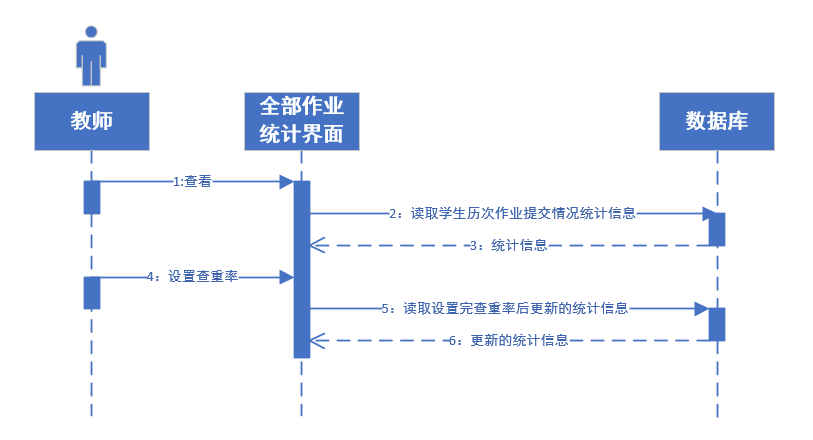


图5 教师查看作业信息统计顺序图

**3. 系统总体设计**

**3.1系统体系结构设计**

系统采用MVC设计模式（即model、view、controller），分别存放数据库模型（php）、前端文件（html）、后端控制文件（php）。如图6所示



图6 系统体系结构图

**3.2 设计问题域子系统**

EmptyClass、Checkuser、Bbs、File、Homework、Index、Lesson、Login、MOSS、Myanswer、Myquestion、ResourceModel、Sharefile、ShareFileRecord、Teacher、User这些是Controller文件夹里的类，负责后端的数据控制。

Bbs\_answerModel、Bbs\_questionModel、Homework\_submitModel、HomeworkModel、Lession\_applyModel、Lession\_userModel、LessionModel、ResourceModel、ShareModel、statistic\_dowloadModal、statistic\_previewModal、UserModel这些是Model文件夹里的类，负责连接数据库。

该系统符合需求分析时所做的类图。

**3.3 设计数据管理子系统**

1. homework为作业信息表：存储着各项作业的作业号、创建作业的教师用户号、课程号、作业题目、作业内容、起止时间、创建时间、作业文件等信息。具体表结构见表1所示。

表1 作业表（homework）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **中文含义** | **字段名** | **类型** | **长度** | **备注** | **完整性约束** |
| 作业号 | id | int | 10 | - | 主键 |
| 用户编号 | userID | int | 10 | 布置作业的教师编号 | 非空 |
| 课程号 | lessonID | int | 10 | 布置作业的课程编号 | 非空 |
| 作业题目 | title | varchar | 128 | 作业的题目名称 | 非空 |
| 作业内容 | content | text | 0 | - | - |
| 开始时间 | starttime | datetime | 0 | 作业可以开始提交时间 | - |
| 结束时间 | endtime | datetime | 0 | 作业截止提交时间 | - |
| 创建时间 | createtime | datetime | 0 | - | 非空 |
| 作业文件 | file\_MD5 | varchar | 64 | 教师附加说明文件，转储为MD5格式 | - |
| 文件名 | filename | varchar | 128 | 作业文件名 | - |
| 知识点号 | KnowledgePointID | int | 10 | - | - |
| 演示样例1 | test\_case | varchar | 64 | 用于解释输入样例形式1 | - |
| 预期结果1 | correct\_result | varchar | 64 | 输入样例对应结果1 | - |
| 演示样例2 | test\_case1 | varchar | 64 | 用于解释输入样例形式2 | - |
| 预期结果2 | correct\_result1 | varchar | 64 | 输入样例对应结果2 | - |

2. homework\_submit为作业提交信息表：存储着每个学生为每次作业提交的作业的信息。具体表结构见表2所示。

表2 提交作业表（homework\_submit）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **属性名** | **字段名** | **类型** | **长度** | **备注** | **完整性约束** |
| 作业文件号 | id | int | 10 | - | 主键 |
| 用户编号 | userID | int | 10 | 提交作业的学生编号 | 非空 |
| 作业号 | homeworkID | int | 10 | 作业的编号 | 非空 |
| 成绩 | score | int | 3 | - | - |
| 文件名 | filename | varchar | 128 | 作业文件名 | - |
| 作业文件 | file\_MD5 | varchar | 64 | 转储为MD5格式 | - |
| 创建时间 | createtime | datetime | 0 | - | 非空 |

3. user为用户信息表：存储着各个用户的信息（包括教师和学生两种）。具体表结构见表3所示。

表3 用户表（user）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **属性名** | **字段名** | **类型** | **长度** | **备注** | **完整性约束** |
| 用户编号 | ID | int | 10 | 用户的编号 | 主键 |
| 姓名 | name | varchar | 32 | 用户姓名 | - |
| 手机号 | phone | varchar | 11 | 用户手机号 | - |
| 用户证件号 | CardId | varchar | 32 | 用户证件号 | - |
| 用户类型 | type | enum | 0 | - | student或teacher |
| 邮件 | email | varchar | 64 | 用户的邮箱地址 | - |
| 性别 | sex | enum | 0 | - | 值为“男”或“女” |
| 学校 | school | varchar | 128 | 用户所在学校名 | - |
| 学院 | academy | varchar | 128 | 用户所在学院名 | - |
| 班级 | major | varchar | 32 | 用户所在班级 | - |
| 学号 | studentID | varchar | 32 | 学生学号 | - |
| 用户名 | username | varchar | 32 | 用户用户名 | 非空 |
| 密码 | password | varchar | 64 | 用户登录密码 | 非空 |
| 最后登录时间 | LastLogInTime | datatime | 0 | 用户最后一次登录的日期、时间 | - |

4.KnowledgePoint为知识点信息表：存储着每个知识点的编号，知识点所属课程的额编号，知识点名称等信息。具体表结构见表4所示。

表4 知识点表（KnowledgePoint）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **中文含义** | **字段名** | **类型** | **长度** | **备注** | **完整性约束** |
| 编号 | id | int | 10 | 自动编号 | 主键 |
| 课程编号 | lessonID | int | 10 | - | - |
| 知识点名称 | name | varchar | 64 | - | - |

5．lesson为课程信息表：存储各个课程相关信息，包括课程编号，课程名称，该课程任课教师的用户编号，创建时间等。具体表结构见表5所示。

表5 课程表（lesson）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **中文含义** | **字段名** | **类型** | **长度** | **备注** | **完整性约束** |
| 课程编号 | ID | int | 10 | - | 主键 |
| 课程名 | lessonName | varchar | 64 | - | 非空 |
| 教师编号 | masterID | int | 10 | 该课程任课教师的编号 | 非空 |
| 创建时间 | createTime | int | 10 | 创建该课程的日期、时间 | 非空 |

6. lesson\_user为课程用户对应信息表：存储着用户及课程之间的对应关系信息。具体表结构见表6所示。

表6 课程用户关系表（lesson\_user）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **中文含义** | **字段名** | **类型** | **长度** | **备注** | **完整性约束** |
| 申请编号 | id | int | 10 | 成功申请的申请号 | 主键 |
| 用户编号 | userID | int | 10 | 该申请的学生编号 | 非空，外键 |
| 课程编号 | lessonID | int | 10 | 该申请的课程编号 | 非空，外键 |

7.lesson\_apply为未审核的学生申请信息表：存储着待审核的课程申请信息。具体表结构见表7所示。

表7 课程申请表（lesson\_apply）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **中文含义** | **字段名** | **类型** | **长度** | **备注** | **完整性约束** |
| 申请编号 | id | int | 10 | 学生申请课程的申请号 | 主键 |
| 用户编号 | userID | int | 10 | 申请课程的学生编号 | 非空，外键 |
| 课程编号 | lessonID | int | 10 | 被申请的课程编号 | 非空，外键 |

8. resource为资源信息表，包含作业MD5文件，作业的类型，存储地址，创建时间等信息。具体表结构见表8所示。

表8 资源信息表（resource）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **属性名** | **存储代码** | **类型** | **长度** | **备注** | **完整性约束** |
| 作业文件 | MD5 | varchar | 64 | 作业转储为MD5格式 | 主键 |
| 作业类型 | type | enum | - | 作业类型（文件或视频） | 非空 |
| 存储地址 | address | varchar | 512 | 文件存储的地址 | 非空 |
| 创建时间 | createTime | datetime | - | - | 非空 |

**3.3 设计人机交互子系统**

系统登录界面如图6 所示，用户需要输入自己的用户名、密码。系统获取用户输入的信息后进入数据库对用户信息进行验证后方可进入系统。若需注册可点击注册按钮，提交相应信息即可完成注册。



图6 登录界面

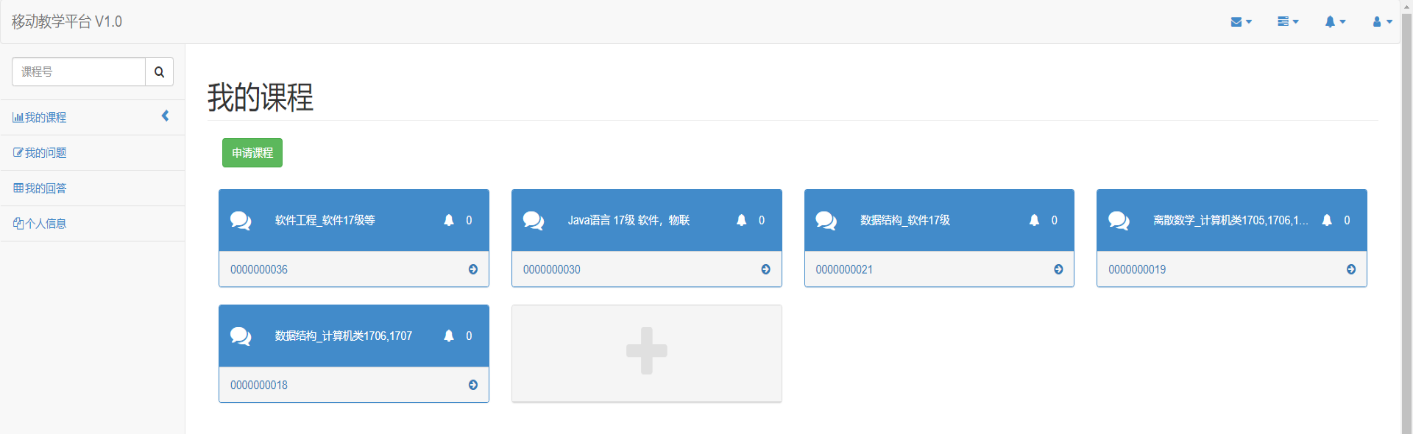
用户主界面如图7所示，主界面主要由课程栏、问题栏、回答栏、个人信息栏组成。且系统右上角会给出消息、设置、个人信息提示等查看按钮。

图7用户主界面

课程信息主界面如图8所示，新增功能：学生可以下载已提交的作业（如果学生未提交本次作业，则右方的下载按钮则不会出现）。



图8课程信息主界面

教师查看作业提交情况的界面如图9所示，在这个界面教师可以查看作业提交情况，同时可以在右上角设置要求的最低重复率（默认50%），再点查重按钮后，便可显示本次作业的名单，详见图10

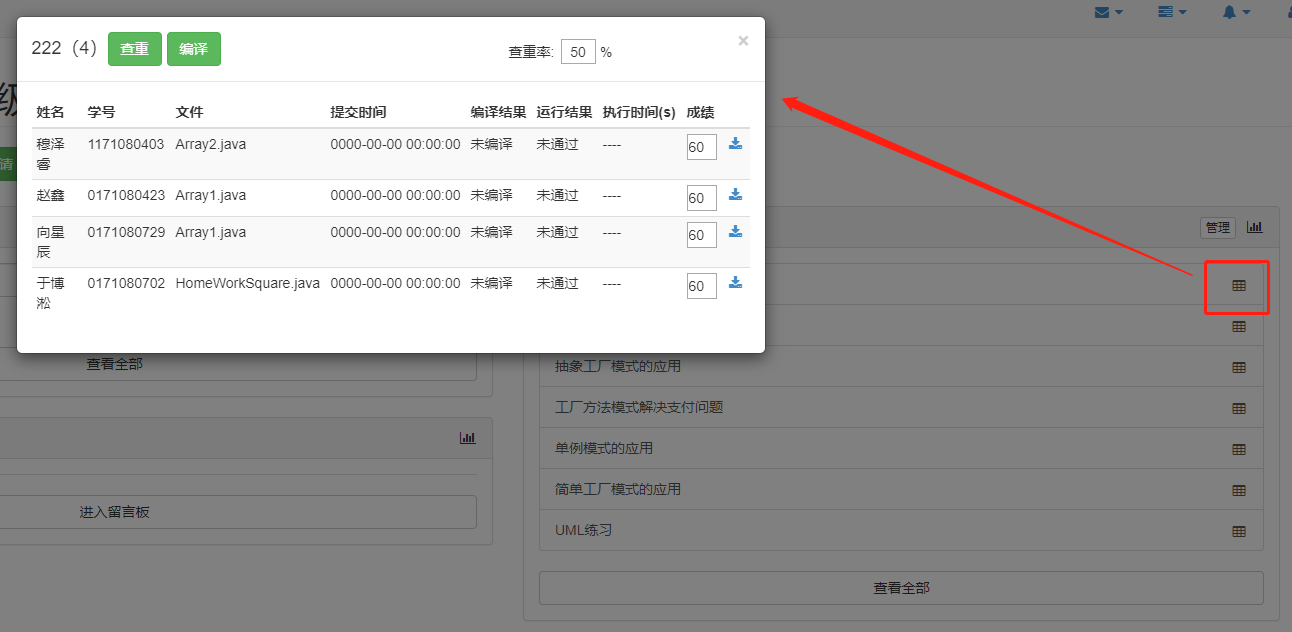


图9作业详情界面

如图10所示，重复率会显示三种名单：①未检查名单：在名单的同学，因提交的文件不符合格式要求（如：\*.rar，\*.zip），未进入检查名单。②完全相同名单：这样的学生完全就是交了同一份文件，为防止提交给moss后，出现太多重复的名单，我们将其单独拎出，并将其在数据库中的chachong值置为100，意为完全重复。③查重名单：剩余的学生作业提交给moss，moss将返还一些意思抄袭的匹配记录，若查到的相似度超过老师一开始定的值，则会被置于这个名单。



图10重复率检查界面

教师查看作业统计情况的界面如图11所示，在这个界面教师可以查看本门课程所有作业的提交及查重情况，同时可以在右上角设置要求的最低重复率（默认60%），再点查看按钮后，便可显示本次作业的名单。



图11作业统计界面

**4. 系统详细设计与实现**

**4.1查重模块**

主要后端函数：Teacher.php中的ajax\_chachong()函数。

登录界面如图8所示，教师用户输入最低限度重复率，系统获取用户输入的信息后连接后台数据库中的算法、调用moss，等待moss返回结果后显示最终结果，最终结果如图9。

**4.1.1获得前端数据并做前期准备工作**

获得当前作业编号，老师设定的查重率，并检验参数是否正确传入。

$homeworkID = input('post.homeworkID'); //当前作业编号

$chachong\_rate = input('post.chachong\_rate'); //获得老师设定的查重率

$data = [];

$data['code'] = 0;

if(empty($homeworkID)){

$data['code'] = 1023;

$data['errmsg'] = '缺少必要参数';

return $data;

}

// 将该作业所有提交文档查重率为0

$res = Homework\_submitModel::getInstance()->setAllChachongto0($homeworkID);

if ($res < 0) {

$data['code'] = 10;

$data['data'] = '数据库设置查重为0时报错';

return $data;

}

$homeworklist = Homework::getInstance()->searchByCondition(['id'=>$homeworkID]);

if(empty($homeworklist)){

$data['code'] = 1232;

$data['errmsg'] = '没有此项作业';

return $data;

}

$submitList = Homework::getInstance()->

searchByCondition\_homework\_submit(['homeworkID' => $homeworkID]);

//得到当前作业所有文件记录

if(!is\_array($submitList)){

$data['code'] = 1024;

$data['errmsg'] = $submitList;

return $data;

}

if(empty($submitList)){

$data['code'] = 0;

$data['data'] = $submitList;

return $data;

}

$errList = [];

$userIDs = [];

$file\_MD5s=[];

$ids = [];

foreach ($submitList as $v){

$filenamev = $v['filename'];

$filenamev = pathinfo($filenamev)['extension'];

if($filenamev == "zip" || $filenamev == "rar" || $filenamev == "pages")

array\_push($errList, $v['userID']);

else{

array\_push($ids, $v['id']);

array\_push($userIDs,$v['userID']);

array\_push($file\_MD5s, $v['file\_MD5']);

}

}

if(empty($errList))

$data['lose'] = $errList;

else{

$con = [];

$con['ID'] = ['op' => 'in','val'=>$errList];

$erruserList = User::getInstance()->searchByCondition($con);

$userIDss=[];

foreach ($erruserList as $v){

array\_push($userIDss,$v['name']);

}

$data['lose'] = $userIDss;

}

**4.1.2将符合规范的待检测文件传给Stanford\_Moss进行查重**

使用申请到的斯坦福MOSS系统的mossid将文件发送给对方服务器进行查重，将其返回的Result网页进行信息抓取，选出符合要求的数据保存在数组中。

// 使用Stanford\_MOSS进行查重

// 抓取result网页中所需信息，包括：组数、比例、行数、MD5文件

function matchInContent($content) {

$DetailInList\_TotalNum = "(right)";

$DetailInList\_Percent = "(\d\*%)";

$DetailInList\_LineNum = "#(?<=(right>))[0-9]\*#";

$DetailInList\_MD5 = "([a-zA-Z0-9]{32})";

preg\_match\_all($DetailInList\_TotalNum, $content, $tmpNum); //组数

preg\_match\_all($DetailInList\_Percent, $content, $tmpPer); //各组相同的比例

preg\_match\_all($DetailInList\_LineNum, $content, $tmpLine); //各组相同的行数

preg\_match\_all($DetailInList\_MD5, $content, $tmpMD5); //各组MD5文件

$result['grpNum'] = sizeof($tmpNum[0]);

$result['percent'] = $tmpPer[0];

$result['line'] = $tmpLine[0];

$result['file\_md5'] = $tmpMD5[0];

return $result;

}

// 根据老师设置的单次作业重复率阈值显示数据

function showDetailsOverChachongRate($details, $rate) { //全部配对结果和预设的阈值

$res = []; // 记录所有大于阈值的数据

$res['grpNum'] = 0; // 记录数据的组数

for($i=0; $i<$details['grpNum']; $i++) {

$perStr1=substr($details['percent'][$i\*2],0,

strlen($details['percent'][$i\*2])-1);

$perStr2=substr($details['percent'][$i\*2+1],

strlen($details['percent'][$i\*2+1])-1);

$perNum1 = number\_format($perStr1); // 第i组第1个文件的重复率

$perNum2 = number\_format($perStr2); // 第i组第2个文件的重复率

$rateNum = number\_format($rate); // 老师设置的重复率阈值

// 如果2个文件有一个超过阈值，即加入数据

// 单选按钮，|| 还是 &&

if($perNum1 > $rateNum && $perNum2 > $rateNum) {

$res['grpNum'] = $res['grpNum'] + 1;

$res['percent'][] = $details['percent'][$i\*2];

$res['percent'][] = $details['percent'][$i\*2 + 1];

$res['line'][] = $details['line'][$i];

$res['file\_md5'][] = $details['file\_md5'][$i\*2];

$res['file\_md5'][] = $details['file\_md5'][$i\*2 + 1];

$res['userid'][] = $details['userid'][$i\*2];

$res['userid'][] = $details['userid'][$i\*2 + 1];

$res['username'][] = $details['username'][$i\*2];

$res['username'][] = $details['username'][$i\*2 + 1];

$res['matchNum'][] = $i;

}

}

return $res;

}

$filetest = "D:/xampp/htdocs/ydjxpt/public/uploads/file/"; //待查文件路径

$mossid = "592439032"; //Stanford\_MOSS用户ID，建议使用者重新发邮件申请

$moss = new MOSS($mossid);

// $moss->addBaseFile('D:\xampp\htdocs\MOSS-PHP-master\Example.java');

//可选项：添加示例文档

// $moss->setCommentString("test1.2");//可选项：添加comment

// 将所有不重复文件加入查重范围

$fileAddList = []; //记录上传给MOSS的文件

$UserIDAddList = []; //记录上传给MOSS的文件对应的UserID

$UserIDWithSamefile = []; //记录提交了相同作业的数组

foreach ($submitList as $sub) {

$flag = 0;

$UserIDWithSamefile[$sub['file\_MD5']][] = $sub['userID'];

// 跳过后缀为zip, rar, pages的文件

$fileExtension = pathinfo($sub['filename'])['extension'];

if($fileExtension == "zip" || $fileExtension == "rar" || $fileExtension == "pages")

continue;

// 查看是否有相同文件已被上传

foreach ($fileAddList as $a) {

if($sub['file\_MD5'] == $a) {

$flag = 1;

break;

}

}

// 若未在已上传文件中找到相同文件，则提交给MOSS并记录

if($flag == 0) {

$addFileName = $filetest.$sub['file\_MD5'];

$moss->addFile("$addFileName");

array\_push($fileAddList, $sub['file\_MD5']);

}

}

**4.1.3统计交同一份文件的学生信息**

获得交完全相同的文件的学生信息，包括MD5（之前获得）, userID , userName，以方便于前端展示以及修改数据库中homework\_submit表中的chachong值。

// 将使用同一份文件的userID转换为userName并记录

$data['UserNameWithSameFileMD5'] = [];

$data['SameID'] = [];

foreach ($UserIDWithSamefile as $list) {

if(count($list) > 1) {

$data['SameID'][] = $list;

for($i=0; $i<count($list); $i++) {

$list[$i] = User::getInstance()->getNameByID($list[$i]);

}

$data['UserNameWithSameFileMD5'][] = $list;

}

}

foreach ($fileAddList as $f) {

$id = Homework::getInstance()->getUserIDByMD5andHomeworkID($f, $homeworkID);

array\_push($UserIDAddList, $id);

}

$data['addlist'] = $fileAddList; //上传给MOSS的所有文件

$data['addListUserID'] = $UserIDAddList; //上传给MOSS的文件对应的UserID

$data['loseID'] = $errList; //文件格式错误的userID

//$data['UserIDWithSameFile']= Homework::getInstance()->

getUserIDWithSameFile($homeworkID); //得到提交同一份文件的学生名单

$data['moss'] = $moss->send(); //将数据提交给MOSS并返回result网页

$data['moss'] = substr($data['moss'],0,strlen($data['moss'])-1);

//返回的result网页最后有换行符，需删除

$url = $data['moss'];

$content = file\_get\_contents($url); //得到result网页源代码

$details = matchInContent($content); //得到所需信息 并 存入data['details']

$data['details'] = $details;

// 使用MOSS查重结果，将MD5文件转换为userID 和 userName 并记录

for($i=0; $i<$data['details']['grpNum']\*2; $i++) {

$usrID = Homework::getInstance()->

getUserIDByMD5andHomeworkID($data['details']['file\_md5'][$i],$homeworkID); //通过MD5文件得到userID

$usrName = User::getInstance()->getNameByID($usrID); //通过userID得到userName

$data['details']['userid'][] = $usrID;

$data['details']['username'][] = $usrName;

}

**4.1.4修改数据库中的信息**

将交完全相同作业的两个人查重率设为100，将格式不对的查重率置为-1，再根据MOSS返回信息依次记录查重率。

$datain = []; // 记录数据库更新的查重率

if(sizeof($data['SameID']) > 0) {

$sameid = [];

foreach ($data['SameID'] as $ids) {

foreach ($ids as $i) {

$sameid[] = $i;

}

}

$datain = Homework\_submitModel::getInstance()->

changeChachongto100($sameid, $datain);

}

if(sizeof($data['loseID']) > 0) {

$datain = Homework\_submitModel::getInstance()->

changeChachongto\_1($data['loseID'], $datain);

}

if($data['details']['grpNum'] != 0) {

$percent = [];

foreach($data['details']['percent'] as $per){

$percent[] = (int)(substr($per, 0, strlen($per)-1));

}

$datain = Homework\_submitModel::getInstance()->

changeChachongtoPercent($data['details']['userid'], $percent, $datain);

}

if(empty($datain)){

$data['code'] = 3;

$data['errmsg'] = 'datain为空';

} else {

foreach($datain as $id=>$chachong) {

$res= Homework\_submitModel::getInstance()->

updateOneChachong($id, $chachong, $homeworkID);

if ($res === true) {

$data['code'] = 0;

$data['errmsg'] = '更新成功';

continue;

} else {

$data['code'] = 2;

$data['errmsg'] = $res;

}

}

}

$data['datain'] = $datain;

$data['detailsOverRate']=showDetailsOverChachongRate

($data['details'], $chachong\_rate); // 得到超过作业重复率阈值的数据

return $data;

**4.2学生下载模块**

主功能的界面如图7所示，下载界面在前端的代码如下：

{if $usertype === 'student'}

if(data[k]['StudentFileID']!=0){

str2="<td><a class=\'dowload\_file\' style=\"float: right;margin-right: 10px;

margin-top: 5px;\" target=\'\_blank\' href='\_\_URL\_\_/index/sharefile/dowloadFile?id="+

data[k]['StudentFileID'][0]['id']+ "&type=Submit\' style=\'cursor:pointer;\'>

<i class=\'fa fa-download fa-fw\'></i></ a></td>"

}

{/if}

**4.3数据统计模块**

主功能的界面如图10所示，下载界面在后端的代码如下 :

public function ajax\_getHomeworkStatisticList\_2020()

{

$userID = session('userID');

$lessionID = input('post.lessionID');

$chachong = input('post.chachong\_rate');

$data = [];

if (empty($userID) || empty($lessionID)) {

$data['code'] = 1023;

$data['errmsg'] = '缺少必要参数';

return $data;

}

if(Lession::getInstance()->checkIfUserInLession($userID, $lessionID) !== true) {

$data['code'] = 1033;

$data['errmsg'] = '您不在此课程中';

return $data;

}

// 先找到选修了这门课的所有的学生信息

// 通过lessionID来找到选择了这门课程的所有学生，

//并且按照用户类型排序，分别为：master->teacher->student

$users = Lession::getInstance()->getUserListByLessionID($lessionID, true);

if (! is\_array($users)) {

$data['code'] = 1034;

$data['errmsg'] = $users;

return $data;

}

$students = [];

foreach ($users as $k => $v) {

if ($v['type'] === 'teacher' || $v['type'] === 'master') {

continue;

}

$arr = [];

$arr['name'] = $v['name'];

$arr['ID'] = $v['ID'];//不显示

$arr['studentID'] = $v['studentID'];//不显示—>显示

$arr['avescore'] = 0;

$arr['homeworkNum'] = 0;

$arr['submitNum'] = 0;

$arr['copyNum'] = 0;

$arr['bianyi\_correct'] = Homework\_submitModel::getInstance()->

getBianyiCorrectNum($v['ID']);

$arr['run\_correct'] = Homework\_submitModel::getInstance()->

getRunCorrectNum($v['ID']);

array\_push($students, $arr);

}

if (empty($students)) { //如果这门课程没有学生选课的话，$students就是一个空数组！

$data['code'] = 0;

$data['data'] = $students;

return $data;

}

// $homeworkList里面存储了查出来的属于这门课的所有作业信息

$homeworkList = Homework::getInstance()->searchByCondition([

'lessionID' => $lessionID

]);

if (empty($homeworkList)) {

$data['code'] = 0;

$data['data'] = $students;

// 如果这门课没有布置作业的话，就简单的把学生的信息列出来。

return $data;

}

$homeworkNum = count($homeworkList);

foreach ($students as $k => $v) {

$students[$k]['homeworkNum'] = $homeworkNum;

// 每个学生都有相同的被布置的作业总数

}

$homeworkIDs = [];

foreach ($homeworkList as $v) {

array\_push($homeworkIDs, $v['id']);

}

$con = [];

$con['homeworkID'] = [

'op' => 'in',

'val' => $homeworkIDs

];

// 'userID,count(\*),sum(score)'赋值给之后查询条件中的field,

// 这样在查询中就变成'select userID,count(\*),sum(score)'了,其中score是字段值。

// 'userID'赋值给之后查询条件中的group，用于分组查询

$homeworkSubmitList=Homework::getInstance()->searchByCondition\_homework\_submit

($con,null,'userID,count(\*),sum(score)','userID');

if (empty($homeworkSubmitList)) {

// 如果这门课没有人提交作业的话，就简单的把学生信息列出来，

// 初始化的学生信息中的成绩信息为初始化值，提交作业次数是0

$data['code'] = 0;

$data['data'] = $students;

return $data;

}

//如果该课程布置过作业

//用userID作为key值

$homeworkSubmitList2 = [];

foreach ($homeworkSubmitList as $v) {

$homeworkSubmitList2[$v['userID']] = $v;

// 这样就是用userID来代替key值，

// 我们在之后的代码中可以直接使用userID来指代一维数组

}

foreach ($students as $k => $v) {

if (! isset($homeworkSubmitList2[$v['ID']])) {

// 如果!isset()，说明这个学生一次作业也没有交过，

//退出循环，此学生的成绩和提交作业次数使用初始默认值。

continue;

}

$students[$k]['avescore'] = round

($homeworkSubmitList2[$v['ID']]['sum(score)'] / $homeworkNum, 2);

// round函数用来保留小数点后两位

$students[$k]['submitNum'] = $homeworkSubmitList2[$v['ID']]['count(\*)'];

$students[$k]['copyNum'] = Homework::getInstance()->

getCopyNum($v['ID'], $lessionID, $chachong);

}

$data['code'] = 0;

$data['data'] = $students;

return $data;

}

**三、课程设计总结或结论**

1. 环境搭建

1.1 开发框架

我们接手的是老师的项目，原开发人员使用thinkPHP5.0框架。初接手该项目时我们曾尝试使用最新的稳定版本thinkPHP6.0进行开发，与原开发人员交流后我们发现修改框架所需工作量过大，考虑到时间因素我们沿用5.0版本。

1.2 PHP+mysql+Apache

当前最新版本的配置为PHP7+mysql8+Apache2.4，在配置完全部环境后发现原始代码无法执行，与原开发人员交流后与他们最初的配置保持一致，使用XAMPP集成配置环境.

1. 数据库设计

由于原始系统已有一部分数据库，我们在原来的基础上进行补充完善以及再设计，结合系统功能需要增加了一部分关系用于新增功能的实现以及原有功能的优化。但是由于文档不全，原有数据库设计理念的理解给我们带来了很大的麻烦，经过数据验证、功能的对比验证、咨询原开发人员等我们整理好了原有的数据库情况，才能在此基础上进行改进。

1. 时间安排与前期规划

3.1 时间安排

课设进行中我们由于时间规划不当有过大量的时间浪费，这种情况给我们带来了很多困扰，通过此次课设我们对于时间的掌握和规划能力更上一筹，相信这份能力对我们以后的发展有着极大的帮助。未来的开发之路上我们需要更加科学的时间规划。

3.2 前期规划

我们遇到的一大部分问题是由于没有适合的规划以及详细的文档指导，原开发人员与我们距离较远，交流也不太方便。并且从搭建环境开始我们就吃到了没有文档指导的苦头，搭建了不合适的环境以及原始数据库了解困难让我们前期寸步难行。

在后来的开发过程中我们也有一段时间没有找到合适的开发步骤而浪费了很多时间来阅读与我们的工作无关的代码。经过本次课设我们了解到的真实的开发场景应该如何进行原有项目的快速了解，并且亲身体验到了文档的重要性。在我们完成该系统的所有功能后我们会把相关的文字材料整理出来保存好，为以后有人再次进行完善和修改留下良好的基础。

与其同时我们书写了一份帮助别人快速上手此系统的文档，其中记录了我们遇到的困难以及解决方法，防止后人再淌我们走过的混水，希望对以后的系统维护人员有所帮助。

**四、参考文献**

[1] 郑人杰.马素霞.软件工程概论.北京：机械工业出版社，2009.11

[2] 高博等. PHP+MySQL+AJAX Web开发给力起飞. 北京：电子工业出版社，2011.9

[3] KevinYank[澳]. PHP和MySQL Web开发从新手到高手（第五版）. 北京：人民邮电出版社， 2013.8