哈爾濱工業大學（威海）

Harbin Institute of Technology at Weihai

课程设计报告

课程名称：	信息安全设计与实践 III
设计题目：	在线考试系统
院　　系：	计算机科学与技术学院
班　　级：	1604202
姓　　名：	张瑞淇
学　　号：	160700225
指导教师：	周广禄
设计时间：	2019 年 9 月 9 日--2019 年 9 月 20 日

哈尔滨工业大学（威海）

二零一九年九月

一、 课题概述

实现一个供多用户同时使用的在线考试系统，实现对考生、题库、试卷等数据对象的管理操作，根据考生的考试结果自动给出成绩并记录。

二、 系统架构

1. 项目环境

Web 服务器	Apache2.4.39
后台编程语言	PHP5.6.9
数据库	MySQL8.0.12
前台 UI 框架	Bootstrap4.0.0
前台图表插件	ECharts

2. 文件目录

网站前后台功能分离，/back 文件夹下为后台功能代码（教师端），/front 文件夹下为前台功能代码（学生端），/include 文件夹为需要包含的公共文件，/static 文件夹下为引用的静态文件，/根文件夹下包含 index 与 install 两个页面，分别为主页与安装页面，首次访问主页时自动安装，并导入样例数据。

3. 项目地址

https://github.com/RyQcan/online_exam

三、 系统设计

1. Models 数据对象

表名	含义
question	题库。每个客观题为一个记录，由教师管理。包含试题号、试题类型、试题难度、题干、四个选项、正确选项、被选中标记等字段。
quiz	试卷。每个试卷为一个记录，根据教师制定的规则自动生成。包含试卷号、试卷难度、题目数量、每题分数等字段。
quiz_record	考试记录。考生开始考试时自动生成。包含经过 md5 的考生号、试卷号、得分、开始考试时间、考试正常结束时间、提交标记等字段。
rule	组卷规则。由教师管理。包含规则号、难度、题数、每题分数等字段。
student	学生。由教师管理。包含学号、姓名、性别、班级、院系等字段。
users	教师（管理员）。系统默认。包含 id、用户名、密码等字段。

各表的具体结构如图 1 所示

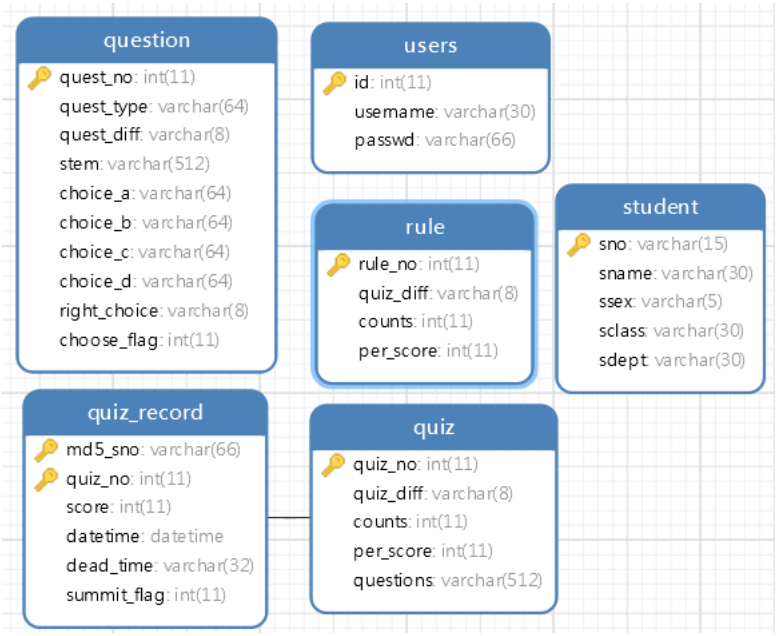

图 1 各表结构

2．View 视图

后台 Controller 接受 Requests 请求，处理 Models 数据，发送 Response 至浏览器，由 Bootstrap 渲染出 UI 页面。

四、 主要功能模块与界面

1．自动安装与样例插入

系统具备首次访问主页时的自动安装（建立数据库、建立数据表）与样例插入的功能。开始使用之前需修改/include 文件夹下的 settings 文件，填写数据库链接相关信息，然后访问主页，系统自动判断是否存在需要的数据库与数据表，若不存在，则自动建立上述信息并自动跳转到前台主页。

代码如下：

```
// 创建数据库
$sql = "CREATE DATABASE " . $dbname;
if ($conn->query($sql) === true) {
    echo "Database created successfully";
    $conn2 = new mysqli($servername, $username, $password, $dbname);
    // 检测连接
    if ($conn2->connect_error) {
        die("Connection failed: " . $conn2->connect_error);
    }
    ...
if ($conn2->multi_query($sql10) && $conn2->multi_query($sql11) &&
$conn2->multi_query($sql12) &&
```

```
    $conn2->multi_query($sql13) && $conn2->multi_query($sql14) &&
$conn2->multi_query($sql15)
    && $conn2->query($sql20) && $conn2->query($sql21) && $conn2->query($sql22)
&& $conn2->query($sql23)) {
    echo "insert successfully";
} else {
    echo "Error insert: " . $conn2->error;
}
$conn2->close();
//安装完成,跳转到主页
    echo "正在跳转到<a href='./front/index.php'>主页</a>";
    header("refresh:1;url=./front/index.php");
} else {
    // 安装失败/重复安装
    echo "Error creating database: " . $conn->error;
}
```

2．管理考生

教师能够对每一名考生进行有效管理，在系统中能够添加、修改、验证以及删除相关考生的信息，在进行敏感操作前检验管理员身份，根据前台传参的 action 参数跳转到对应的代码执行相应功能，由/back/sut.php 完成，主要是对 student 数据表的操作。

导航栏-考生管理-添加考生，添加功能如图 2 所示

图 2 添加考生

导航栏-考生管理-查询考生，查询功能如图 3 所示

图 3 查询考生

根据查询到的考生，点击其右侧修改按钮可以进行信息修改，如图 4 所示

图 4 修改考生

删除功能类似不再赘述

3．管理考试题库

考试题库由全部题目组成，根据每个题目的属性可以方便地进行分类，如科目、难度等，教师可以在后台对试题进行管理，根据前台传参的 action 参数跳转到对应的代码执行相应功能，由/back/questions.php 完成，主要是对 question 数据表的操作。

导航栏-考题管理-添加考题，添加功能如图 5 所示

图 5 添加考题

导航栏-考题管理-查询考题，查询功能如图 6 所示

图 6 查询考题

同样地，点击右侧修改与删除按钮可以实现对应操作，不再赘述。

4．管理在线考试

教师可以编制出题规则，设置试卷难易程度、题目数量、每题分数等，在考生点击开始考试时根据这个规则，系统自动组织试卷。由/back/quiz.php 完成，主要是对 rule 数据表的操作。

导航栏-考试规则，可以对规则进行修改，如图 7 所示

图 7 查看/修改出题规则

5．组卷

当考生点击开始考试时，系统根据教师制定的规则自动组卷，并立即生成一条考试记录，记录题号、考试开始时间、正常结束时间，并设置提交 summit_flag 为 0，表示考生并未提交试卷。

在组卷过程中，由 rule 表中的规则按一定难度比例在试题库中随机抽选 choose_flag 为 0（表示此题目未正被他人选中）题目，并将被选中的题目的 choose_flag 设为 1，表示此题正被选中，若此时有另一考生同时开始考试请求组卷，则系统不会分配相同题目给另一考生，待组卷结束往 quiz_record 数据表插入一条考试记录时，将所有被选中题目的 choose_flag 恢复为 0，可供他人选中。

6．在线考试功能

考虑到在考试中可能存在断网、不小心关闭页面、电脑崩溃等意外情况，在考生每次点击开始考试时，系统自动在 quiz_record 库中查询是否存在此考生并未提交且未超时的试卷，若存在，则根据考试记录的题号，从题库中重新选出这些题目，等待考生作答，由于存储了考试正常结束时间，所以考试倒计时功能可以按照结束时间继续倒计时，系统时间取自服务器时间，所以不会根据前台时间而改动。

若考生关闭页面时间太久，超过考试结束时间，则在考生下次访问开始考试页面时自动提示考生时间已结束，并记录分数为 0。

若考生不存在未提交的试卷，则根据上述功能，自动生成新试卷，并立即开始倒计时，且在离考试结束 5 分钟时提示考生尽快作答，并在数秒后消失。

当考生访问开始考试页面时，自动生成的新试卷如图 8 所示

图 8 自动生成的新试卷

此时 quiz_record 表与 quiz 表都添加了一条新的记录，前者通过试卷编号这一外键在后者中确定被抽中题目的编号。

在 quiz_record 表中生成的考试记录如图 9 所示

md5_sno	quiz_no	score	datetime	dead_time	summit_flag
6e15d6e039825dfc815142c82be04847	1	5	2019-09-11 03:03:01	1568185741	1
6e15d6e039825dfc815142c82be04847	2	20	2019-09-11 03:03:30	1568185770	1
6e15d6e039825dfc815142c82be04847	3	0	2019-09-20 08:49:04	1568940849	1
6e15d6e039825dfc815142c82be04847	4	2	2019-09-20 08:54:43	1568941188	1
6e15d6e039825dfc815142c82be04847	5	0	2019-09-20 09:23:40	1568942925	1
6e15d6e039825dfc815142c82be04847	6	0	2019-09-20 09:24:20	1568942964	1
6e15d6e039825dfc815142c82be04847	7	0	2019-09-22 04:55:35	1569142839	0

图 9 考试对应的考试记录

其中第 7 套试卷在 quiz 表中的数据如图 10 所示

quiz_no	quiz_diff	counts	per_score	questions
1	难	10	5	7*9*24*25*29*32*36*38*40*41*
2	难	10	5	6*8*22*23*28*31*33*35*39*40*
3	易	10	2	3*10*11*12*17*18*25*26*33*38*
4	易	10	2	3*4*7*11*13*17*19*23*26*31*
5	易	10	2	3*8*12*13*18*19*23*24*33*38*
6	易	10	2	3*4*5*14*15*16*18*22*24*36*
7	易	10	2	4*9*14*15*17*18*23*25*29*42*

图 10 考试记录对应的试卷

系统提醒考生考试快结束，如图 11 所示

图 11 提醒考生考试快结束

当考生关闭页面并重新进入后，恢复的考试与倒计时如图 12 所示

图 12 重新打开页面继续作答与倒计时

考生因离开页面超过考试结束时间被判 0 分，如图 13 所示

图 13 关闭页面导致超时

7．自动给出成绩并记录考试

当考生提交试卷后，系统根据对应的题号在 question 表中查询对应的正确答案，并给出考生成绩，将成绩记录在 quiz_record 中，并设置 summit_flag 为 1，表示考试结束，此考生已经提交过这一试卷。防止考生进行重放攻击（试卷已经提交，抓包重放不会被接受）或暴力遍历正确选项（已经提交了试卷，成绩不会再改变）

在考试时间耗尽时，系统强制提交试卷并计算成绩

考生提交试卷时，系统给出成绩，如图 14 所示

图 14 提交试卷给出成绩

8．基于角色的访问控制

由于本系统仅仅涉及两个角色（教师与学生），所以将功能分为前后台，各自的权限与功能在前后台分别实现互不干扰，实现了基于角色的访问控制。

五、 安全考虑

1．解决 SQL 注入攻击

在所有涉及数据库查询的操作中，使用 PHP 参数化查询，有效地解决了 SQL 注入攻击。部分代码如下所示

```
$sql = "INSERT INTO student VALUES(?,?,?,?,?)";
$stmt = $conn->stmt_init();

if ($stmt->prepare($sql)) {
    $stmt->bind_param("sssss", $_GET['sno'], $_GET['sname'], $_GET['ssex'],
$_GET['sclass'], $_GET['sdept']);
    if ($stmt->execute())
• • •
```

2．防止假冒他人身份考试

在插入新的考试记录时，将当前考生的学号加 salt（盐值，一个随机字符串）经过 md5 后存储，即在 quiz_record 表中插入的是加密后的考生号，这样一来，考生通过抓包并不能伪装成他人考试。

3．防止更改试题

在计算分数时，先将前台传来的（试题号-答案）中的试题号与 quiz 表中存储的试题号对比，内容、顺序必须一致的情况下才判断答案是否正确，考生无法通过抓包修改题目编号，使所有题目都为考生会做的那一道题。

4．防止在提交试卷阶段进行重放攻击与暴力遍历正确答案

前面已经讲到，通过设置 summit_flag，表明试卷已经被提交，考试已经结束，不会再对同一考生提交的同一试卷的答案进行判断，分数也不会再改变。

六、 感悟

在实际开发应用的过程中，系统往往会出现令开发人员意想不到的 BUG 与漏洞，这些漏洞轻则打乱业务的正常逻辑，重则会使服务器权限被攻击者获取。这就要求开发人员不仅需要站在用户的角度看待问题，还需要站在攻击者的角度看待问题，思考系统存在哪些漏洞并在开发中避免这些情况出现。

在开发考试系统的过程中，有很多细节上的安全问题，尤其是如何检验考试者的身份，如何防止试题被更改等。核心在于不要信任任何前台传递过来的参数。对前台传来的参数进行过滤，防止 XSS 攻击、SQL 注入攻击；还要对前台参数进行验证，确认发送方身份、是否是正常业务逻辑中产生的数据、是否存在重复提交等。网络安全是一门很深的学问，这就要求我们不断学习、不断进步，善于思考、善于总结。