考试系统项目工程

开发文档

创建日期：2019.09.02

项目4考试系统开发小组

**项目名称:考试系统**

**要求：**

1. 角色：　学生（考试），老师（出题）
2. 老师出题(出题页面)：　试题：考试范围（第一章，　第二章．．．．），每一章都有题库, 题库需要录入, 每次考试, 从每个章节中随机抽出一定数量的题目, 组成考卷. 每份试卷包含所有章节的随机题目.
3. 学生考试(考试页面) – 试卷随机, 试题随机, 考试时间有规定, 设定超时!!!!超时处理: 直接交卷
4. 改卷: 全都是单选题!, 交卷后, 自动判分, 当场给出分数
5. 限制: 记录学生的成绩, 不能重复考.

项目组成员名单:

|  |  |  |
| --- | --- | --- |
| 角色 | 组员 | 备注 |
| 项目经理 | 高原 |  |
| 后端 | 高原, 黄河源 |  |
| 前端 | 李春朋, 满春发 |  |
| UI | 苏壮晖 |  |
| 文档/测试 | 李旭昊, 高原 |  |
|  |  |  |

目 录

[一、用户需求 1](#_Toc18598357)

[1.使用范围: 1](#_Toc18598358)

[2.角色: 1](#_Toc18598359)

[3.目的: 1](#_Toc18598360)

[4.功能需求表 1](#_Toc18598361)

[二、需求分析 2](#_Toc18598362)

[1.系统功能介绍： 2](#_Toc18598363)

[2.系统结构图 2](#_Toc18598364)

[3.模块划分 2](#_Toc18598365)

[3.1系统管理模块 2](#_Toc18598366)

[3.2登录模块 2](#_Toc18598367)

[3.3题库管理模块 3](#_Toc18598368)

[3.4考试/分数管理模块 3](#_Toc18598369)

[3.5 E-R图 3](#_Toc18598370)

[3.6整体功能实现流程图 4](#_Toc18598371)

[三、详细设计 5](#_Toc18598372)

[1数据库表的设计 5](#_Toc18598373)

[1.1创建考试系统库 6](#_Toc18598374)

[1.2创建管理员表(教师登录) 6](#_Toc18598375)

[1.3创建学生表 6](#_Toc18598376)

[2创建题库表(html题库表,js题库表,jq题库表) 7](#_Toc18598377)

[2.1创建html题库表 7](#_Toc18598378)

[2.2 创建js题库表 7](#_Toc18598379)

[2.3 创建jq题库表 7](#_Toc18598380)

[3数据字典: 7](#_Toc18598381)

[3.1名称:学生信息 7](#_Toc18598382)

[3.2名称:老师信息 8](#_Toc18598383)

[3.3名称:试卷信息 8](#_Toc18598384)

[3.4名称:试题信息 8](#_Toc18598385)

[3.5名称:分数信息 8](#_Toc18598386)

[四、项目功能测试 9](#_Toc18598387)

[1.测试环境 9](#_Toc18598388)

[2.功能测试 9](#_Toc18598389)

[2.1各模块功能测试 9](#_Toc18598390)

[2.2前端数据与数据库数据传输测试。 9](#_Toc18598391)

[2.3数据对比测试。 9](#_Toc18598392)

[2.4代码强度测试。 9](#_Toc18598393)

[3.bug记录 10](#_Toc18598394)

[4.解决方案 10](#_Toc18598395)

# 一、用户需求

1.使用范围:各个学校考试测试使用。

2.角色:学生(考试), 老师(出题)。

3.目的:为了让老师可以从题库中随机选取一定数量的题目,组成试卷,试卷中要包含所有章节的随机题目,可以自动评分,减少老师负担.方便老师和学生。

## 4.功能需求表

|  |  |  |
| --- | --- | --- |
| 功能 | 具体描述 | 角色 |
| 登录系统 | 用户输入教职工编号密码登录,学生需要用学号注册,才可以登录 | 学生,老师 |
| 试题管理 | 题库类型分3类,题目为单选题,题目按章节类型存放,  由且只由老师向题库中章节中添加题目, | 老师 |
| 试卷管理 | 学生通过考试页面来随机选取题目,每份试卷包含所有章节的随机题目. | 学生 |
| 考试管理 | 考试时间有规定,超时后出现提交按钮. | 老师 |
| 查看分数 | 提交试卷,确认提交后,确认分数后,学生和老师可以在库里查看已经得出的分数 | 学生,老师 |
| 在线考试 | 学生登录后点击开始考试,在规定时间内开始答题,点击提交试卷后,页面上显示当前分数,点击确认后,结束考试,注销并返回到登录注册页面. | 学生 |
| 条件限制 | 每个学生提交试卷,得出分数后,不可再次答题 | 学生 |

# 二、需求分析

## 1.系统功能介绍：

这个系统主要目的是老师通过在题库中选取章节,规定时间,学生随机获得试题,学生通过登录系统,在规定时间内完成考试,得出分数,学生和老师都可以查看已经提交的试卷;学生得出分数后不可再次考试答题.系统自动组成试卷,并且跟数据库对比完成试卷的判分和统计,来判断题目是否正确。

## 2.系统结构图

在线考试系统

考试/分数管理

登录/注册模块

题库管理模块

显示分数

考试题目随机

自动得出分数

教师添加试题

学生注册模块

老师登录模块

学生登录模块

## **3.模块划分**

### **3.1系统管理模块**

建立考题数据库和学生成绩数据库,用来存放试题和学生成绩,建立相对应的表,在表单里存放题库。

### **3.2登录模块**

学生和老师需要通过登录页面进入在线考试系统.学生在登录之前,首先需要通过登录页面进行注册。

3.2.1老师登录模块:

老师通过教师ID和密码登录。

3.2.2学生注册模块:

学生可进入到注册页面进行注册输入个人信息,点击注册按钮来注册学生用户账号。

3.2.3学生登录模块:

学生可以用学号注册,通过学号和密码登录考试页面. 学生有分数后将不能再次登录考试。

### 3.3题库管理模块

教师添加试题:

教师可在教师页面进行个人出题，做好的试题输入试题库。

### 3.4考试/分数管理模块

1. 考试题目随机：

考生在进入试题界面后所给出的题目出自数据库所有章节的随机选出的题目。

2.自动得出分数

在考生交卷后，系统会将所提交的试题与试题库的正确答案作对比，如果匹配正确则加分,否则不得分。

3.显示分数

最后统计在学生点击提交或者时间结束,得分情况在页面上显示。

### 3.5 E-R图

题目编写

老师

试题

题目推送?

考生

### 3.6整体功能实现流程图

开始

开始

注册账户/登录页面

老师登录

**N**

**Y**

登录成功？

老师出题

学生考试页面

学生成绩页面

出题成功，题目入库

可查看考试题库

考试()

题目从题库随机抽取,遍历章节,在所有章节中随机取出定量题目

学生可查看个人成绩

题目在调用时随机推送至学生考试页面

是否超时？

**Y**

强制提交试卷

结束

N

学生点击提交

判分()

结束

# 三、详细设计

## 1数据库表的设计

|  |
| --- |
| 数据库表名 |
| question |
| students |
| teacher |

学生信息表

|  |  |  |
| --- | --- | --- |
| 数据表名 | 字段名 | 数据类型 |
| students | id | int(11) |
| username | varchar(40) |
| userpassword | varchar(40) |
| userscore | varchar(20) |

教师信息表

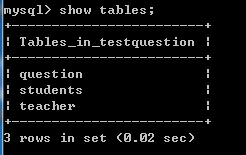
|  |  |  |
| --- | --- | --- |
| teacher | id | int(11) |
| username | varchar(40) |
| userpassword | varchar(40) |

题库信息表

|  |  |  |
| --- | --- | --- |
| 数据表名 | id | int(11) |
| testtitle | varchar(100) |
| optionsA | varchar(50) |
| optionsB | varchar(50) |
| question | optionsC | varchar(50) |
| optionsD | varchar(50) |
| answer | varchar(20) |
| kind | varchar(40) |

### 1.1创建考试系统库

create database testquestion;



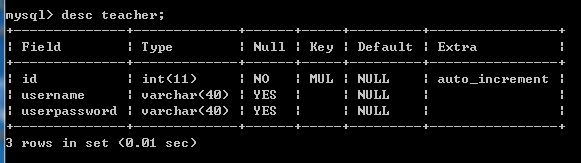
### 1.2创建管理员表(教师登录)

create table teacher (id int auto\_increment,username varchar(40),userpassword varchar(40),key(id));

insert into teacher(username,userpassword) values('user','123456');

管理员表信息

作用:用来保存老师的基本信息,如图:

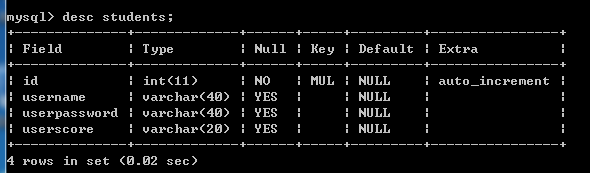


### 1.3创建学生表

create table students(id int auto\_increment,username varchar(40),userpassword varchar(40),userscore varchar(20),key(id));

学生信息表

作用:用来保存学生的注册登录的基本信息,如同:



## 2创建题库表(html题库表,js题库表,jq题库表)

### 2.1创建html题库表

create table htmlquestion(id int auto\_increment,testtitle varchar(100),optionsA varchar(50),optionsB varchar(50),optionsC varchar(50),optionsD varchar(50),answer varchar(20),key(id));

### 2.2 创建js题库表

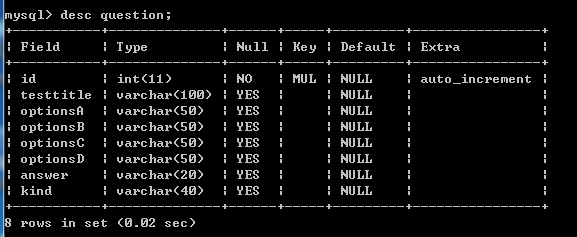
create table javascriptquestion(id int auto\_increment,testtitle varchar(100),optionsA varchar(50),optionsB varchar(50),optionsC varchar(50),optionsD varchar(50),answer varchar(20),key(id));

### 2.3 创建jq题库表

create table jqueryquestion(id int auto\_increment,testtitle varchar(100),optionsA varchar(50),optionsB varchar(50),optionsC varchar(50),optionsD varchar(50),answer varchar(20),key(id));

题库信息

作用:用于保存老师向各个章节中写入考试题目的信息,如图:



## 3数据字典:

### 3.1名称:学生信息

简述:包含学生学号和密码,用来注册登录考试页面

数据流去向:输入到数据库中

### 3.2名称:老师信息

简述:包含老师的职工编号和密码,用来登录管理页面

数据流去向:输入到数据库中

### 3.3名称:试卷信息

简述:由老师向题库中输入题目,学生来随机选取组成

数据流来源:由学生随机选取试题组成

数据流去向:供学生考试

### 3.4名称:试题信息

简述:由老师向三个章节输入题目,全部为单选题

数据流来源:由老师输入获得

数据流去向:为试卷供题

### 3.5名称:分数信息

简述:学生提交试卷后自动判分获得

数据流来源:自动对比数据库得出分数

数据流去向:数据库中保存以及显示在本次提交试卷的学生页面上

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据表名 | 字段名 | 数据类型 | 是否为空 | 备注 |
| students | id | int(11) | no | id |
| username | varchar(40) | yes | 学生姓名 |
| userpassword | varchar(40) | yes | 密码 |
| userscore | varchar(20) | yes | 分数 |
| question | id | int(11) | no | id |
| testtitle | varchar(100) | yes | 题目 |
| optionsA | varchar(50) | yes | 选择A |
| optionsB | varchar(50) | yes | 选择B |
| optionsC | varchar(50) | yes | 选择C |
| optionsD | varchar(50) | yes | 选择D |
| answer | varchar(20) | yes | 答案 |
| kind | varchar(40) | yes | 种类 |
| teacher | id | int(11) | no | 职工编号 |
| username | varchar(40) | yes | 教师姓名 |
| userpassword | varchar(40) | yes | 密码 |

# 四、项目功能测试

## 1.测试环境

1. wamp自带SQL数据库。
2. 谷歌浏览器。

## 2.功能测试

### 2.1各模块功能测试

测试方案：按照登录注册以及编题考试流程进行测试，测试各个模块是否能实现该功能。

测试结果：功能正常运行，页面转换流畅。

### 2.2前端数据与数据库数据传输测试。

测试方案：在页面输入相对应信息后，查看数据库是否成功接收；在数据库表单建立相对应数据表，页面能否成功输出。

测试结果：产生bug-e，在用户名和密码输入带有空格时也能成功登陆;产生bug-f，若输入的ID和密码匹配选项都不存在，则会导致数据库崩溃。

### 2.3数据对比测试。

测试方案：在页面或者数据库输入几组不同数据，对比其产生结果的不同。

测试结果：产生bug-a,bug-d，使得前端数据与数据库输出数据出错；已修复。

### 2.4代码强度测试。

测试方案：在页面查看是否有报错，逻辑是否一致，数据是否完整，以及在不同的浏览器下的兼容。

测试结果：出现bug-b，bug-c，已修复；浏览器兼容性需要客户注意版本问题，以及项目功能的兼容更新。

## 3.bug记录

a.初期建表时，编码设置不正确，导致输入数据库的中文不识别，导入导出均为问号。

b.引用的jQuery为网络地址，而引用的bootstrap.js为本地地址，导可能因为网络原因，bootstrap.js无法找到和依赖jQuery，异步执行导致报错。

c.把连接数据库语句变成外部模块的时候，connection没有在全局定义，而是写在模块里，导致其他模块无法调用。

d.注册界面输入空信息也可以进行注册登录并且参加考试。

e.在用户名和密码输入带有空格时也能成功登陆。

f. 若输入的ID和密码匹配选项都不存在，则会导致数据库崩溃。

## 4.解决方案

解决a：在建表语句后加上DEFAULT CHARSET=utf8;

解决b：把jQuery引用和bootstrap.js引用都变为本地地址引用。

解决c：在全局定义。

解决d：在注册的ajax容错里添加拦截代码，拦截空信息并谭弹出注册失败弹窗。

解决e：通过输入的用户名和密码与数据库进行匹配，匹配成功才能登录。

解决f：在登录模块进行容错。