



软件工程课程设计

测试报告

目录

1. 测试目的	3
2. 测试环境	3
3. 测试环节	3
3.1 问题概述及解决方案	3
4. 建议及结论	5
4.1 测试改进建议	5
4.2 测试结论	5

1. 测试目的

软件测试时软件开发过程中一个非常重要的组成部分。是对软件产品进行验证和确认的过程。在软件投入运行前，对软件需求分析、设计规格说明和编码的最终复审，是软件质量保证的关键步骤。其目的是尽早尽快地发现在软件产品存在的问题，检查软件中的 bug。写成软件测试报告交于开发人员进行修改。我们需要制定每个测试阶段测试完成以及测试成功的标准、要实现的目标。通过对系统的每个功能进行白盒测试或者黑盒测试，识别出测试活动中各种风险，并消除可能存在的风险，降低由不可能消除的风险所带来的损失。

2. 测试环境

在浏览器端进行测试。

3. 测试环节

3.1 问题概述及解决方案

在距离项目截止期前三天，我们整合完成了课程注册系统。此后，我们展开了密集的测试工作。

经过测试，本小组共找到了 28 个问题，列举一些典型问题如下：

问题编号	问题概述	解决方案
1	如果学生栏输入的毕业年份小于出生年月日也会存入数据库	在更新数据库之前加入安全判断
2	在地址栏直接输入界面时会直接进入该界面，安全性低	在每个页面加载前加入判断，如果超全局变量\$_SESSION 中未记录用户则会跳转到登陆界面
3	登陆界面“记住密码”复选框第一次取消时仍会自动输入密码	更改自动输入密码判断
4	教授界面显示已选课程人数不正确	将 flag=0 设置为 sql 语句的条件之一
5	教授界面已选课人数与课程打分界面显示不同步	将 flag=0 设置为 sql 语句的条件之一
6	已选课程中会显示空信息课程但选课人数不为 0	设计数据库时 sec_id 与 takes 中不统一，更改后解决
7	学生已选课程中有冲突课程	数据库初始化时数据冲突已存在，修改后解决
8	教授选课界面会出现不同学院的课程	更改教授课程列表 sql 语句条件
9	教授选课界面中会出现不同学年不同学	在数据库中增加当前学年和学期

	期的课程	信息
10	教授选课界面中会出现不同学年不同学期的课程	在数据库中增加当前学年和学期信息
11	教授选择添加时间冲突课程仍能添加成功	数据库修改后 sql 语句未能及时更新，更新 sql 语句后解决
12	登陆界面用户名和密码输入错误后会将已输入的信息清空	使用 ajax 技术局部刷新页面
13	教授课程选择后已选课程仍在列表中	教授成功选课后刷新页面
14	学生课程选择后已选课程仍在列表中	教授成功选课后刷新页面
15	学生备选课程和正式课程时间冲突仍可选择	当正选和备选冲突时不允许选课
16	导航栏显示的当前选项不正确	更改当前标签中的 active 属性
17	导航栏中用户名显示为用户 ID	使用 sql 语句查询到 ID 对应的用户名后进行显示
18	链接跳转不正确	将文件从 html 修改为 php 后，对对应的跳转链接也加以修改
19	整合之后页面跳转错误	将各个功能的同名文件改名
20	当数据库关闭时没有提示信息	当数据库连接失败时弹窗提示
21	总是显示有用户正在注册（管理员自身）	管理员登录时不将个人在线信息添加到数据库
22	修改密码时新旧密码相同仍显示修改密码成功	当新旧密码相同时，弹窗提示新旧密码不能相同，不做数据库修改
23	个人信息中密码会以明码的形式显示，安全性低	将密码设置为暗文
24	传送密码时网页后缀会跟上用户名和密码，安全性低	将提交模式从 get 改为 post
25	管理员搜索教授 ID 时，输入学生 ID 也可成功进入下一界面	限制只能输入 P 开头的字符串，格式不正确时弹窗提示
26	管理员搜索学生 ID 时，输入教授 ID 也可成功进入下一界面	限制只能输入 S 开头的字符串，格式不正确时弹窗提示
27	系统关闭时教授仍可选课	Sql 语句中加上对系统关闭的判断
28	系统关闭时学生仍可选课	Sql 语句中加上对系统关闭的判断
29	点击按钮时无反应	Php 文件中有错误，进行改正
30	会有功能之外的弹窗提示	将测试用的 alert 语句去除
31	系统关闭后学生备选没有被正确的加入到主选中	修改 sql 语句，对属性值加上双引号
32	系统关闭后，课程已被删除但 teaches 信息还存在	增加新的删除语句
33	系统关闭后，课程的删除操作没有正常进行	将三句 sql 语句分开执行
34	课程表中不能正确的显示时间	修改 sql 语句中的条件
35	数据库连接失败时弹窗提示无效	更改弹窗语句的位置
36	教授个人信息用户名未显示	更改显示用户名的语句

4. 建议及结论

4.1 测试改进建议

本次测试过程一共经历了五次的黑盒测试和三次白盒测试,对系统的交互、功能、接口、逻辑等进行了全方位的测试,对软件需求分析、设计规格说明和编码的最终复审。经过测试与修正,本系统的可用性、易用性、安全性、稳定性都已经达到了预期的要求。各方面的功能较系统刚刚完成时也得到了较大的拓展。

4.2 测试结论

1. 在测试过程中,由于起初设计的测试用例不全面而导致一些问题不能得到及时的发现,在进行测试时要注意设计好全面详尽的测试计划,才能全面的对系统进行测试,找到系统的缺陷所在。
2. 在测试中最好是由模块的实现者来测试本模块。
3. 组内成员及时沟通尤为重要。尤其是及时更新版本进度、更新补丁文档、实时分享任务进度。