DocNo: 001. H.1:1

Code Cafe

测试计划书

Version 2.0

**By:**

Cold Cafe

2012-12

**Group Member:**

谷宇昊

林亦洋

洪彦

王烺

**Document Language:**

中文

版本历史

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <11/12/2012> | <1.0> | Code Café详细的测试文档 | 王烺 |
| <01/02/2013> | <2.0> | Code Café详细的测试文档 | 洪彦 |
|  |  |  |  |
| **Final date** | **<x.x>** | **<details>** | **<name>** |

目录

[1 介绍 5](#_Toc343175641)

[1.1目的 5](#_Toc343175642)

[1.2背景 5](#_Toc343175643)

[1.3参考书 5](#_Toc343175644)

[2计划 5](#_Toc343175645)

[2.1计划概述 5](#_Toc343175646)

[2.2测试用例 6](#_Toc343175647)

[2.3单元测试 7](#_Toc343175648)

[2.3.1测试时间表 7](#_Toc343175649)

[2.3.2条件 7](#_Toc343175650)

[2.3.3测试参考 7](#_Toc343175651)

[2.3.4测试训练 8](#_Toc343175652)

[2.4集成测试 8](#_Toc343175653)

[2.4.1测试时间表 8](#_Toc343175654)

[2.4.2条件 8](#_Toc343175655)

[2.4.3测试参考 9](#_Toc343175656)

[2.4.4测试训练 9](#_Toc343175657)

[2.5系统功能测试 9](#_Toc343175658)

[2.5.1测试时间表 9](#_Toc343175659)

[2.5.2条件 9](#_Toc343175660)

[2.5.3测试参考 9](#_Toc343175661)

[2.5.4测试训练 9](#_Toc343175662)

[2.6运行时测试 9](#_Toc343175663)

[2.6.1测试时间表 10](#_Toc343175664)

[2.6.2条件 10](#_Toc343175665)

[2.6.3测试参考 10](#_Toc343175666)

[2.6.4测试训练 10](#_Toc343175667)

[2.7压力测试 10](#_Toc343175668)

[2.7.1测试时间表 10](#_Toc343175669)

[2.7.2条件 10](#_Toc343175670)

[2.7.3测试参考 10](#_Toc343175671)

[2.7.4测试训练 11](#_Toc343175672)

[3测试设计细节 11](#_Toc343175673)

[3.1单元测试 11](#_Toc343175674)

[3.1.1控制方法 11](#_Toc343175675)

[3.1.2输入 11](#_Toc343175676)

[3.1.3输出 11](#_Toc343175677)

[3.1.4流程 11](#_Toc343175678)

[3.2集成测试 12](#_Toc343175679)

[3.2.1控制方法 12](#_Toc343175680)

[3.2.2输入 12](#_Toc343175681)

[3.2.3输出 12](#_Toc343175682)

[3.2.4流程 12](#_Toc343175683)

[3.3系统功能测试(登陆部分) 12](#_Toc343175684)

[3.3.1控制方法 12](#_Toc343175685)

[3.3.2输入 13](#_Toc343175686)

[3.3.3输出 13](#_Toc343175687)

[3.3.4步骤 13](#_Toc343175688)

[3.4系统功能测试 (注册) 13](#_Toc343175689)

[3.4.1控制方法 13](#_Toc343175690)

[3.4.2输入 14](#_Toc343175691)

[3.4.3输出 14](#_Toc343175692)

[3.4.4步骤 14](#_Toc343175693)

[3.5系统功能测试(登出) 14](#_Toc343175694)

[3.5.1控制方法 14](#_Toc343175695)

[3.5.2输入 14](#_Toc343175696)

[3.5.3输出 14](#_Toc343175697)

[3.5.4过程 14](#_Toc343175698)

[3.6系统功能测试(公告添加) 15](#_Toc343175699)

[3.6.1控制方法 15](#_Toc343175700)

[3.6.2输入 15](#_Toc343175701)

[3.6.3输出 15](#_Toc343175702)

[3.6.4过程 15](#_Toc343175703)

[3.7系统功能测试(修改公告) 15](#_Toc343175704)

[3.7.1控制方法 15](#_Toc343175705)

[3.7.2输入 15](#_Toc343175706)

[3.7.3输出 16](#_Toc343175707)

[3.7.4过程 16](#_Toc343175708)

[3.8系统功能测试(查找公告) 16](#_Toc343175709)

[3.8.1控制方法 16](#_Toc343175710)

[3.8.2输入 16](#_Toc343175711)

[3.8.3输出 16](#_Toc343175712)

[3.8.4过程 16](#_Toc343175713)

[3.9系统功能测试(删除公告) 16](#_Toc343175714)

[3.9.1控制方法 17](#_Toc343175715)

[3.9.2输入 17](#_Toc343175716)

[3.9.3输出 17](#_Toc343175717)

[3.9.4过程 17](#_Toc343175718)

[3.10系统功能测试 (创建群体讨论) 17](#_Toc343175719)

[3.10.1控制方法 17](#_Toc343175720)

[3.10.2输入 17](#_Toc343175721)

[3.10.3输出 17](#_Toc343175722)

[3.10.4过程 18](#_Toc343175723)

[3.11系统功能测试(点对点讨论) 18](#_Toc343175724)

[3.11.1控制方法 18](#_Toc343175725)

[3.11.2输入 18](#_Toc343175726)

[3.11.3输出 18](#_Toc343175727)

[3.11.4过程 18](#_Toc343175728)

[3.12系统功能测试(项目文件创建) 18](#_Toc343175729)

[3.12.1控制方法 18](#_Toc343175730)

[3.12.2输入 19](#_Toc343175731)

[3.12.3输出 19](#_Toc343175732)

[3.12.4过程 19](#_Toc343175733)

[3.13系统功能测试(项目文件下载) 19](#_Toc343175734)

[3.13.1控制方法 19](#_Toc343175735)

[3.13.2输入 19](#_Toc343175736)

[3.13.3输出 19](#_Toc343175737)

[3.13.4过程 19](#_Toc343175738)

[3.14运行时测试 20](#_Toc343175739)

[3.14.1控制方法 20](#_Toc343175740)

[3.14.2输入 20](#_Toc343175741)

[3.14.3输出 20](#_Toc343175742)

[3.14.4过程 20](#_Toc343175743)

[3.15压力测试 20](#_Toc343175744)

[3.15.1控制方法 20](#_Toc343175745)

[3.15.2输入 20](#_Toc343175746)

[3.15.3输出 20](#_Toc343175747)

[3.15.4过程 21](#_Toc343175748)

[4测试标准 21](#_Toc343175749)

[4.1范围 21](#_Toc343175750)

[4.1.1缺陷检出率 21](#_Toc343175751)

[4.1.2覆盖率标准 21](#_Toc343175752)

[4.2数据目录 21](#_Toc343175753)

[4.3测试规模 21](#_Toc343175754)

[4.3.1测试终止标准 21](#_Toc343175755)

[4.3.2单元测试终止标准 22](#_Toc343175756)

[4.3.3集成测试终止标准 22](#_Toc343175757)

[4.3.4系统测试终止标准 22](#_Toc343175758)

# 1 介绍

## 1.1目的

这个文档是CodeCafe系统的测试计划，它包括了测试环境，测试范围和测试标准等。该文档将作为我们测试主要的参考。因此该文档的主要阅读者是CodeCafe的开发人员和测试人员。

## 1.2背景

这个系统叫做CodeCafe，开发者为ColdCafe团队（成员包括：谷宇昊，林亦洋，洪彦，王烺）。CodeCafe是供小型开发团队使用的团队管理与沟通系统。整个项目开始于11月4日。经过了需求分析，系统设计和编写代码，下一个环节就是测试。

## 1.3参考书

《Software Testing》 Ron Patton

# 2计划

## 2.1计划概述

|  |  |  |
| --- | --- | --- |
| 功能 | 输入 | 输出 |
| 登录 | 用户名和密码 | 如用户密码对应，则表示成功；  否则返回失败。 |
| 注册 | 用户名，密码和确认密码 | 如用户名此前不存在且密码与确认密码对应，则返回成功；  否则返回失败。 |
| 登出 | 用户点击登出选项 | 成功登出 |
| 公告添加 | 需要填写公告标题、内容和时间，三项都填写完毕后，可以确认提交。 | 需要填写公告标题、内容和时间，三项都填写完毕后，可以确认提交，返回成功；  否则返回失败。 |
| 修改公告 | 在自己发布的公告列表中选择需要修改的公告，点击后显示已填写的公告标题、内容和时间，可以直接在此基础上进行修改，点击确认提交。 | 如成功，则公告板上变为已修改的公告 |
| 查找公告 | 在搜索栏输入需要查找的关键词或时间。 | 输出为满足搜索输入的公告列表，每页显示10项。 |
| 删除公告 | 在自己发布的公告列表中选择需要删除的公告，点击删除，被选中的公告名,id将作为输入提交。 | 成功删除公告 |
| 创建群体讨论 | 用户可以点击添加群体讨论，然后在名单列表中选择需要邀请的成员加入一起讨论。 | 如果创建成功，输出返回的是新创建讨论的id；  否则返回失败。 |
| 点对点讨论 | 类似于群体讨论，只是讨论只限于两者之间，自己用户id和对方成员id作为输入。 | 如果创建成功，输出返回的是新创建讨论的id；  否则返回失败。 |
| 项目文件创建 | 用户在功能栏点击创建，弹出对话框，然后在地址栏中选择中文件或者文件夹。提交后选中的文件名将作为输入。 | 新的文件被创建成功。 |
| 项目文件下载 | 用户在主界面选择要下载的文件，然后在功能栏点击下载，如果设置了默认下载路径，则默认路径名和下载的文件名作为输入。如果没有设置下载路径，则会弹出对话框，让用户选择路径并确认后，路径名和下载文件名为输入。 | 用户选择的项目文件被下载到用户个人系统中。 |
| 项目文件更新 | 用户在主界面选择要更新的文件，然后在功能栏点击更新，弹出对话框，然后在地址栏中选择文件或者文件夹。 | 创建成功输出返回新项目文件id号，并将原文件更新。 |
| 项目文件删除 | 和更新文件类似，用户在主界面选择要删除的文件，然后在功能栏点击删除，将会弹出确认对话框，用户再次确认后，选中的文件名作为输入。 | 用户选择的文件被删除，并返回删除项目文件的id号。 |

## 2.2测试用例

|  |  |  |
| --- | --- | --- |
| 测试名称 | 测试进程 | 测试内容与目的 |
| 单元测试 | 12.12-12.18 | 使用黑盒测试和白盒测试来检查每个单元，该测试和编写代码一同进行。 |
| 集成测试 | 12.12-12.18 | 通过黑盒测试来检测单元接口和继承功能。 |
| 系统功能测试 | 12.18-12.22 | 使用黑盒测试来验证需求的一致性。需满足所有的功能要求已被实现，所有的性能要求已被满足。 |
| 运行时测试 | 12.23-12.23 | 在运行时刻使用黑盒测试，以检测该软件是否符合需求分析的条件。 |
| 压力测试 | 12.24-12.24 | 检测服务器的可靠性。 |

## 2.3单元测试

所有成员都将参与到这个测试，由于此为功能测试，软件的每一个部分都将被检测。

### 2.3.1测试时间表

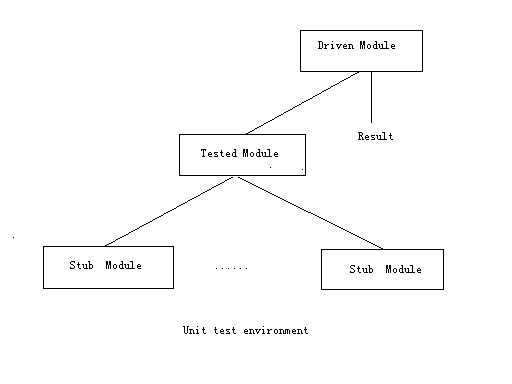
此测试的计划时间为12月12日至12月18日。

### 2.3.2条件

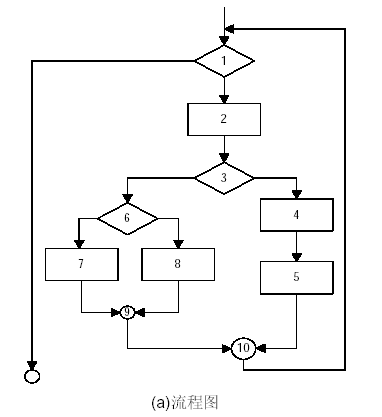
1. 4 台电脑(CPU: P4,Memory:512M),需要5天。
2. 所有成员都将参与到该次测试，我们需要了解黑盒测试与白盒测试。

### 2.3.3测试参考

1. CodeCafe的需求文档.
2. CodeCafe的用户手册
3. 单元测试:



1. 当使用白盒测试时，测试者需要将程序流程转换为流程图



**工作流程图**

### 2.3.4测试训练

无

## 2.4集成测试

所有成员都将参与到这个测试，由于此为功能测试，软件的每一个部分都将被检测。

## 2.4.1测试时间表

此测试的计划时间为12月12日至12月18日。

## 2.4.2条件

a) 4 台电脑(CPU: P4,Memory:512M),需要5天。

b) 所有成员都将参与到该次测试，我们需要了解黑盒测试。

## 2.4.3测试参考

1. CodeCafe的需求文档.
2. 整个CodeCafe代码包括(登陆，注册，登出，公告添加，修改公告，查找公告，删除公告，创建群体讨论，点对点讨论，项目文件创建，项目文件下载，项目文件更新，项目文件删除)
3. 集成测试的环境和流程图与单元测试一致

### 2.4.4测试训练

无

## 2.5系统功能测试

所有成员都将参与到这个测试，由于此为功能测试，软件的每一个部分都将被检测。

## 2.5.1测试时间表

此测试的计划时间为12月18日至12月22日。

## 2.5.2条件

a) 4 台电脑(CPU: P4,Memory:512M),需要5天。

b) 所有成员都将参与到该次测试，我们需要了解黑盒测试。

## 2.5.3测试参考

1. CodeCafe的需求文档.
2. 整个CodeCafe代码包括(登陆，注册，登出，公告添加，修改公告，查找公告，删除公告，创建群体讨论，点对点讨论，项目文件创建，项目文件下载，项目文件更新，项目文件删除)

### 2.5.4测试训练

无

## 2.6运行时测试

所有成员都将参与到这个测试，由于此为功能测试，软件的每一个部分都将被检测。

## 2.6.1测试时间表

此测试的计划时间为12月23日至12月23日。

## 2.6.2条件

a) 4 台电脑(CPU: P4,Memory:512M),需要1天。

b) 所有成员都将参与到该次测试，我们需要了解黑盒测试。

## 2.6.3测试参考

1. CodeCafe的需求文档.
2. 整个CodeCafe代码包括(登陆，注册，登出，公告添加，修改公告，查找公告，删除公告，创建群体讨论，点对点讨论，项目文件创建，项目文件下载，项目文件更新，项目文件删除)

### 2.6.4测试训练

无

## 2.7压力测试

所有成员都将参与到这个测试，由于此为功能测试，软件的每一个部分都将被检测。

## 2.7.1测试时间表

此测试的计划时间为12月24日至12月24日。

## 2.7.2条件

a) 2台电脑(CPU: P4,Memory:512M),需要1天。

b) 所有成员都将参与到该次测试，我们需要了解如何使用WAS。

## 2.7.3测试参考

CodeCafe的需求文档

### 2.7.4测试训练

无

# 3测试设计细节

## 3.1单元测试

是整个测试中最小规模的测试。使用白盒测试来检测单元内部的指令，使用黑盒测试来检测功能和测试目标的动作。单元测试是随着开发一起进行的。

### 3.1.1控制方法

每一个单元都是由测试者手动地完成。因为在我们的开发过程中，开发者就是测试者，他们能够在发现问题的时候马上修复。

### 3.1.2输入

1. 使用白盒测试：测试用例必须保证经过了独立的路径，以验证包括了所有的程序分支。
2. 使用黑盒测试：测试用例必须将输入数据分割成等价类，并从每个等价类中选择测试用例。

### 3.1.3输出

根据相关存根模块，驱动模块和测试脚本，该单位应产生预期的结果。

## 3.1.4流程

1. 设计测试用例，测试文件以test\_[所测试类名].py为形式
2. 编写存根模块，驱动模块和自动化测试脚本run.py，会将本文档下所有test开头形式的测试文件载入
3. 执行代码python run.py，unittest会将结果与期望比较
4. 修改bug，直到所有bug均被修复
5. 当不存在bug时，返回报告，测试结束

## 3.2集成测试

根据系统的业务层，表示层，和子系统，整合相关单位的测试集成版。使用黑盒测试检查的功能和作用的集成版。整个过程采用底部–顶集成。测试人员必须写出正确的存根模块，驱动模块，与测试脚本。

### 3.2.1控制方法

每一个集成模块都是由测试者手动地完成。因为在我们的开发过程中，开发者就是测试者，他们能够在发现问题的时候马上修复。

了解了持续集成的工具：持续集成服务器Hudson以及版本管理工具git和svn

### 3.2.2输入

使用黑盒测试：测试用例必须使用等价类划分和边界设计设计测试用例。此外，查询模型和统计模型，测试人员可以使用因果图。

### 3.2.3输出

根据相关存根模块，驱动模块和测试脚本，该单位应产生预期的结果。

## 3.2.4流程

1. 设计测试用例
2. 编写存根模块，驱动模块和测试脚本，建立一个测试用数据库
3. 运行服务器，执行代码，将结果与期望比较
4. 修改bug，直到所有bug均被修复
5. 当不存在bug时，测试结束

## 3.3系统功能测试(登陆部分)

这个部分由谷宇昊和林亦洋进行测试。目的是检测是否用户具有权限，非法用户应当被拒绝。

### 3.3.1控制方法

手动进行注册新用户，观察注册结果，并记录在Excel中。

### 3.3.2输入

1. 错误的用户ID
2. 错误的密码
3. 无ID输入
4. 无密码输入
5. 过长的用户ID
6. 过长的密码
7. 正确的ID，错误的密码
8. 正确的ID和密码

### 3.3.3输出

1. 显示用户ID错误
2. 显示密码错误
3. 显示ID不能为空
4. 显示密码不能为空
5. 显示用户ID过长
6. 显示密码过长
7. 显示密码错误
8. 成功登陆

## 3.3.4步骤

1. 设置不同的用户名和密码
2. 设计测试用例
3. 运行系统
4. 手动输入测试用例，观察结果
5. 如果所有的测试用例的输出都达到标准，则完成测试

## 3.4系统功能测试 (注册)

这个部分由谷宇昊和林亦洋进行测试。目的是检测是否可以新注册用户。

### 3.4.1控制方法

手动进行注册新用户，观察注册结果，并记录在Excel中。

### 3.4.2输入

新注册的用户名，密码及确认密码

### 3.4.3输出

显示新用户注册成功

## 3.4.4步骤

1. 运行系统
2. 手动进行新用户的注册
3. 观察注册的结果是否符合标准

## 3.5系统功能测试(登出)

这个部分由谷宇昊和林亦洋进行测试。目的是检测是否让用户顺利地退出系统。

### 3.5.1控制方法

手动将用户退出系统，观察退出，并记录在Excel中。

### 3.5.2输入

已登录的用户点击登出选项。

### 3.5.3输出

如用户成功登出，返回1，否则返回0。

## 3.5.4过程

1. 运行系统
2. 手动进行用户的登出
3. 观察结果是否符合标准，如正确则停止测试

## 3.6系统功能测试(公告添加)

This part is tested by Mario and Roger, to check that whether to close a project as the

这个部分由谷宇昊和林亦洋进行测试。目的是检测用户能否顺利地添加公告。

### 3.6.1控制方法

手动添加公告，观察公告板，并记录在Excel中。

### 3.6.2输入

公告标题，内容和时间，最后点击提交。

### 3.6.3输出

公告板显示新的公告。

## 3.6.4过程

1. 运行系统
2. 手动进行公告的添加
3. 观察结果是否符合标准，如正确则停止测试

## 3.7系统功能测试(修改公告)

这个部分由谷宇昊和林亦洋进行测试。目的是检测用户能否顺利地修改公告。

### 3.7.1控制方法

手动修改公告，观察公告板，并记录在Excel中。

### 3.7.2输入

选择要修改的公告，点击后显示当前的标题，内容和时间，可以直接在此基础上进行修改。

### 3.7.3输出

公告板显示修改后的公告。

## 3.7.4过程

1. 运行系统
2. 手动进行公告的修改
3. 观察结果是否符合标准，如正确则停止测试

## 3.8系统功能测试(查找公告)

这个部分由谷宇昊和林亦洋进行测试。目的是检测用户能否顺利地查找所需的公告。

### 3.8.1控制方法

手动查找公告，观察查找结果，并记录在Excel中。

### 3.8.2输入

需要查找的关键词或时间。

### 3.8.3输出

输出为满足搜索条件的公告列表，一页显示10项。

## 3.8.4过程

1. 运行系统
2. 创建多条公告
3. 手动进行公告的查找
4. 观察结果是否符合标准，如正确则停止测试

## 3.9系统功能测试(删除公告)

这个部分由谷宇昊和林亦洋进行测试。目的是检测用户能否顺利地删除公告。

### 3.9.1控制方法

手动查找公告，观察查找结果，并记录在Excel中。

### 3.9.2输入

在自己发布的公告列表中选择需要删除的公告，点击删除。

### 3.9.3输出

公告板删除用户选择的公告。

## 3.9.4过程

1. 运行系统
2. 创建多条公告
3. 手动进行公告的删除
4. 观察结果是否符合标准，如正确则停止测试

## 3.10系统功能测试 (创建群体讨论)

这个部分由谷宇昊和林亦洋进行测试。目的是检测用户能否顺利地创建群体讨论。

### 3.10.1控制方法

手动创建群体讨论，观察创建结果并进行讨论，并记录在Excel中。

### 3.10.2输入

用户可以点击添加群体讨论，然后在名单列表中选择需要邀请的成员加入一起讨论。然后在讨论组中输入需要讨论的内容。

### 3.10.3输出

创建讨论成功时显示提示信息，此后讨论组任何一名用户的讨论内容都可以被其他小组成员所见。

## 3.10.4过程

1. 多台电脑同时运行系统
2. 由一名用户创建群体讨论
3. 组内用户发布讨论信息
4. 观察结果是否符合标准，如正确则停止测试

## 3.11系统功能测试(点对点讨论)

这个部分由谷宇昊和林亦洋进行测试。目的是检测用户能否顺利地创建点对点讨论。

### 3.11.1控制方法

手动创建点对点讨论，观察创建结果并进行讨论，并记录在Excel中。

### 3.11.2输入

用户可以点击添加群体讨论，然后在名单列表中选择一名需要邀请的成员加入一起讨论。然后在点对点讨论中输入需要讨论的内容。

### 3.11.3输出

创建讨论成功时显示提示信息，此后讨论组任何一名用户的讨论内容都可以被另一名成员所见。

## 3.11.4过程

1. 多台电脑同时运行系统
2. 由一名用户创建点对点讨论
3. 组内用户发布讨论信息
4. 观察结果是否符合标准，如正确则停止测试

## 3.12系统功能测试(项目文件创建)

这个部分由谷宇昊和林亦洋进行测试。目的是检测用户能否顺利地创建新的项目文件。

### 3.12.1控制方法

手动创建新的项目文件，观察创建文件结果，并记录在Excel中。

### 3.12.2输入

用户在功能栏点击创建，弹出对话框，然后在地址栏中选择中文件或者文件夹。提交后选中的文件名将作为输入。

### 3.12.3输出

文件列表中会显示新创建的文件。

## 3.12.4过程

1. 运行系统
2. 由一名用户创建新的项目文件
3. 观察结果是否符合标准，如正确则停止测试

## 3.13系统功能测试(项目文件下载)

这个部分由谷宇昊和林亦洋进行测试。目的是检测用户能否顺利地下载项目文件。

### 3.13.1控制方法

手动下载项目文件，观察下载文件结果，并记录在Excel中。

### 3.13.2输入

用户在主界面选择要下载的文件，然后在功能栏点击下载，如果设置了默认下载路径，则默认路径名和下载的文件名作为输入。如果没有设置下载路径，则会弹出对话框，让用户选择路径并确认后，路径名和下载文件名为输入。

### 3.13.3输出

用户所选择的文件被顺利下载，并显示下载成功。

## 3.13.4过程

1. 多台电脑同时运行系统
2. 多名用户分别创建不同的项目文件
3. 用户分别下载项目文件
4. 观察结果是否符合标准，如正确则停止测试

## 3.14运行时测试

这部分是测试和系统测试一起进行，检查系统运行时间达到运行标准，即系统响应时间（响应用户请求从服务器到客户端）在网络状态正常的条件下必须小于20秒。

### 3.14.1控制方法

手工测试，将测试结果记录在Excel中。

### 3.14.2输入

每一个模块，必须选择运行时间最长的测试用例，如果用时小于20秒，模块通过标准。

### 3.14.3输出

经过一定的响应时间，系统输出结果。

## 3.14.4过程

1. 设计测试用例
2. 进行测试并记录bug
3. 如果所有测试用时均符合时间标准，则终止测试

## 3.15压力测试

使用WAS进行测试。

### 3.15.1控制方法

使用WAS自动进行压力测试，并且由WAS产生测试结果。

### 3.15.2输入

模拟有100名用户同时在使用这个系统，检测系统的流畅程度。

### 3.15.3输出

WAS会提供对应的测试结果。

## 3.15.4过程

1. 建立一个完整的数据库
2. 设计测试用例
3. 运行服务器
4. 使用WAS进行测试，并记录缺陷
5. 在测试结果符合预想时，结束测试

# 4测试标准

## 4.1范围

测试用例的覆盖率必须达到100%

## 4.1.1缺陷检出率

1. 1类和2类的缺陷必须被全部检测出
2. 3类和4类的缺陷检出率必须达到80%
3. 5类缺陷的检出率必须达到60%

## 4.1.2覆盖率标准

1. 语句的覆盖率必须达到80%
2. 测试用例的覆盖率必须达到100%
3. 需求测试的覆盖率必须达到100%

## 4.2数据目录

测试者手工地将测试结果与期望的结果进行比较，并在Excel中记录bug。

## 4.3测试规模

## 4.3.1测试终止标准

1. 在单元测试，集成测试，系统测试之后，测试都已经达到了相应的标准。
2. 该软件通过了确认试验，并产生了确认实验的结论。

## 4.3.2单元测试终止标准

1. 单元测试用例均已完成
2. 根据测试用例,测试者已经完成了所有的单元测试
3. 达到了单元测试的覆盖率标准
4. 保证每千行代码发现至少3处缺陷
5. 单元函数必须与设计模式一致
6. 所有的缺陷已经被修正，修正率达到标准

## 4.3.3集成测试终止标准

1. 集成测试用例均已完成
2. 根据集成测试用例, 测试者已经完成了所有的集成测试
3. 达到了集成测试的覆盖率标准
4. 保证每千行代码至少发现2处缺陷
5. 集成功能与性能必须与需求文档要义一致
6. 所有的缺陷已经被修正，修正率达到标准

## 4.3.4系统测试终止标准

1. 系统测试用例均已完成
2. 根据系统测试计划, 测试者已经完成了所有的系统测试
3. 达到了系统测试的覆盖率标准
4. 保证每千行代码至少发现1处缺陷
5. 系统功能与性能必须与需求文档要义一致
6. 所有的缺陷已经被修正，修正率达到标准