# (UH}P]$EA(6KVUYP8]L~{{I

# 

# 失物招领平台测试计划

小组：SEG1

成员：苏雨豪

陆凯晨

吴桐

目录

[1简介 3](#_Toc485261577)

[1.1目的 3](#_Toc485261578)

[1.2背景 3](#_Toc485261579)

[2测试参考文档和测试提交文档 3](#_Toc485261580)

[2.1测试参考文档 3](#_Toc485261581)

[2.2测试提交文档 3](#_Toc485261582)

[3测试进度 3](#_Toc485261583)

[4测试资源 4](#_Toc485261584)

[4.1人力资源 4](#_Toc485261585)

[4.2测试环境 4](#_Toc485261586)

[4.3测试工具 4](#_Toc485261587)

[5评价准则 5](#_Toc485261588)

[5.1范围 5](#_Toc485261589)

[5.2测试活动文档 5](#_Toc485261590)

[5.3准则 6](#_Toc485261591)

[6测试方案 6](#_Toc485261592)

[6.1单元测试 6](#_Toc485261593)

[6.2集成测试 11](#_Toc485261594)

[6.3系统测试 11](#_Toc485261595)

### 1简介

#### 1.1目的

本文档为完成《失物招领系统》的各个测试阶段所需测试活动提供计划。

本文档要提供给质量小组作为复查的依据，同时要提供给开发小组。质量小组和开发小组都是SEG1。

#### 1.2背景

2017年下学期软件工程小组G01在完成失物招领系统的代码实现后，开始进行对系统的各项测试。在开始前制定此计划，并以此为指导完成该系统的测试。

### 2测试参考文档和测试提交文档

#### 2.1测试参考 文档

参考文献：

Junit测试的相关文章<http://www.jianshu.com/p/7088822e21a3>

<http://blog.csdn.net/andycpp/article/details/1327147>

#### 2.2测试提交文档

各项测试的输入以及输出。

每项测试的测试报告。

测试人员的分析及建议。

### 3测试进度

|  |  |  |  |
| --- | --- | --- | --- |
| 测试活动 | 计划开始日期 | 实际开始日期 | 结束日期 |
| 制定测试计划 | 2017.5.30 | 2017.6.6 | 2107.6.12 |
| 单元测试 | 2017.6.6 | 2017.6.6 | 2017.6.12 |
| 集成测试 | 2017.6.6 | 2017.6.13 | 2017.6.18 |
| 系统测试 | 2017.6.6 | 2017.6.13 | 2017.6.18 |
| 用户验收测试 | 2017.6.12 | 2017.6.14 | 2017.6.14 |
| 对测试进行评估 | 2017.6.12 | 2017.6.14 | 2017.18 |

### 4测试资源

#### 4.1人力资源

|  |  |
| --- | --- |
| 人员 | 具体职责或注释 |
| 吴桐 | 单元测试 |
| 苏雨豪 | 系统测试 |
| 陆凯晨 | 集成测试 |

#### 4.2测试环境

软件环境（相关软件、操作系统等）：

MySQL数据库、eclipse（包含Junit测试包）、失物招领系统代码

硬件环境（网络、设备等）：

联网的笔记本电脑。各组员的笔记本即可。

#### 4.3测试工具

|  |  |  |  |
| --- | --- | --- | --- |
| 用途 | 工具 | 生产商/自产 | 版本 |
| 单元测试 | Junit | Oracle | Junit4 |
| 集成测试 | Junit | Oracle | Junit4 |
| 系统测试 | Junit | Oracle | Junit4 |

### 5评价准则

#### 5.1范围

单元测试内容：

本系统一共由四个模块构成：用户模块、信息发布模块、搜索模块、删除模块

对用户模块进行单元测试可能发现的问题是：用户账号重复、用户名或密码过长产生未知错误等。这些错误的影响较小，一般情况下不会影响系统运行。但此类错误必须全部避免才能通过用户模块的单元测试。

对信息发布模块进行单元测试可能发现的问题：发布的信息不符合内容的要求等。这类错误对系统的使用影响较大，影响到大量用户的正常浏览。此类错误必须解决，才能通过该模块的单元测试。

对搜索模块地单元测试。一般不会出现问题，关键是SQL语句中的模糊查询语句的正确。必须确保搜索语句的正确性。

对删除模块的单元测试。测试用例必须百分之百确保被删除，但在数据库中会保留记录，属于软删除。所有用例都被执行才能通过该模块的单元测试。

#### 5.2测试活动文档

测试工作产品检查表：

|  |  |  |
| --- | --- | --- |
| 检查项 | 裁减标准 | 检查项目 |
| 测试计划 | 不可裁减 |  |
| 测试大纲 | 不可裁减 |  |
| 测试用例 | 不可裁减 |  |
| 测试用例评审表 | 不可裁减 |  |
| 测试计划评审表 | 可裁减 |  |
| 测试遗留（待改）问题确认表 | 不可裁减 |  |
| 测试问题卡 | 可裁减 |  |
| 测试报告 | 不可裁减 |  |

#### 5.3准则

进入准则：

系统预测试（随机单击某一模块，基本能够完成整个操作，弹出系统崩溃少于2次），通过后再正式进入测试。

退出准则：

·测试通过：

1. 没有严重的缺陷
2. 发现的缺陷在商定的要求之内：如最终确定不在本版本中处理留待进一步的升级版本解决
3. 测试用例的执行百分比100%

·测试没通过或暂时退出：

1. 一些进度标准：如必须终止测试以便按计划交付用户
2. 由于AUT（被测应用系统）不满足这个水平上的测试，测试工作必须终止。也就是说如果AUT的质量不好，进一步的测试工作是浪费。判定本版本AUT暂时不可测。

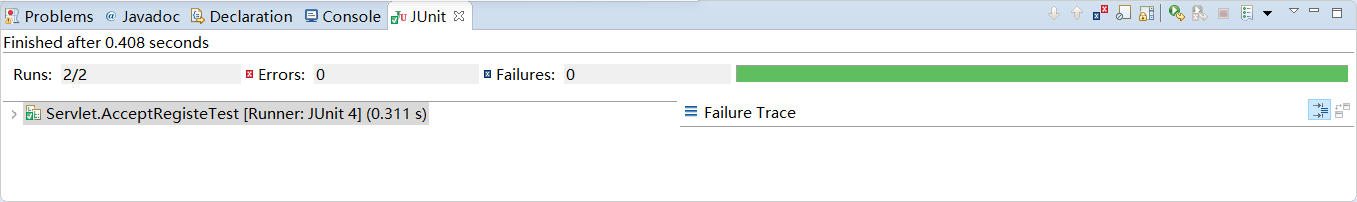
### 6测试方案

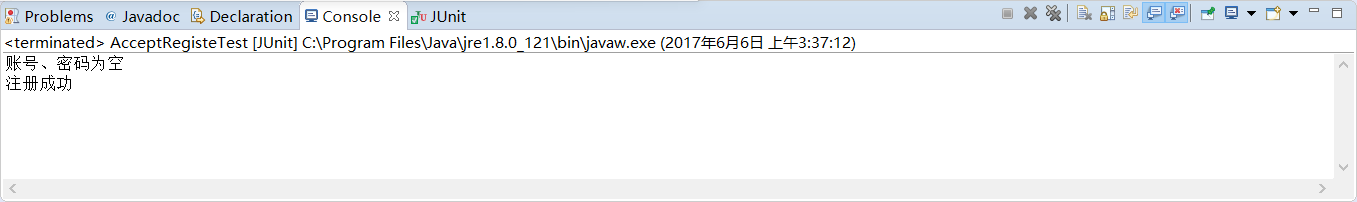
#### 6.1单元测试

1）用户模块

注册测试 register()

|  |  |  |
| --- | --- | --- |
| 操作 | 输入 | 输出 |
| 不输入 | 不输入 | 账号、密码为空 |
| 输入账号、密码 | 账号：31501394  密码：31501394 | 注册成功 |



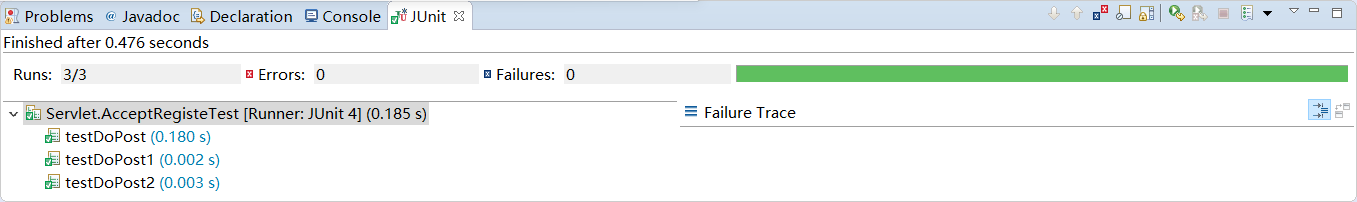




2）判定覆盖

|  |  |  |
| --- | --- | --- |
| 操作 | 输入 | 输出 |
| 两次密码输入不一致 | UsePwd=31501394  UserPwd=3150395 | 两次密码输入不一致 |
| 两次密码输入一致 | UserPwd=UsePwd1 | 注册成功 |

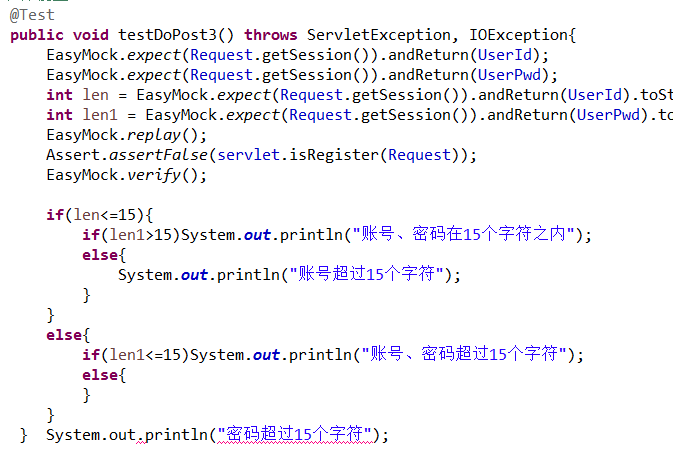




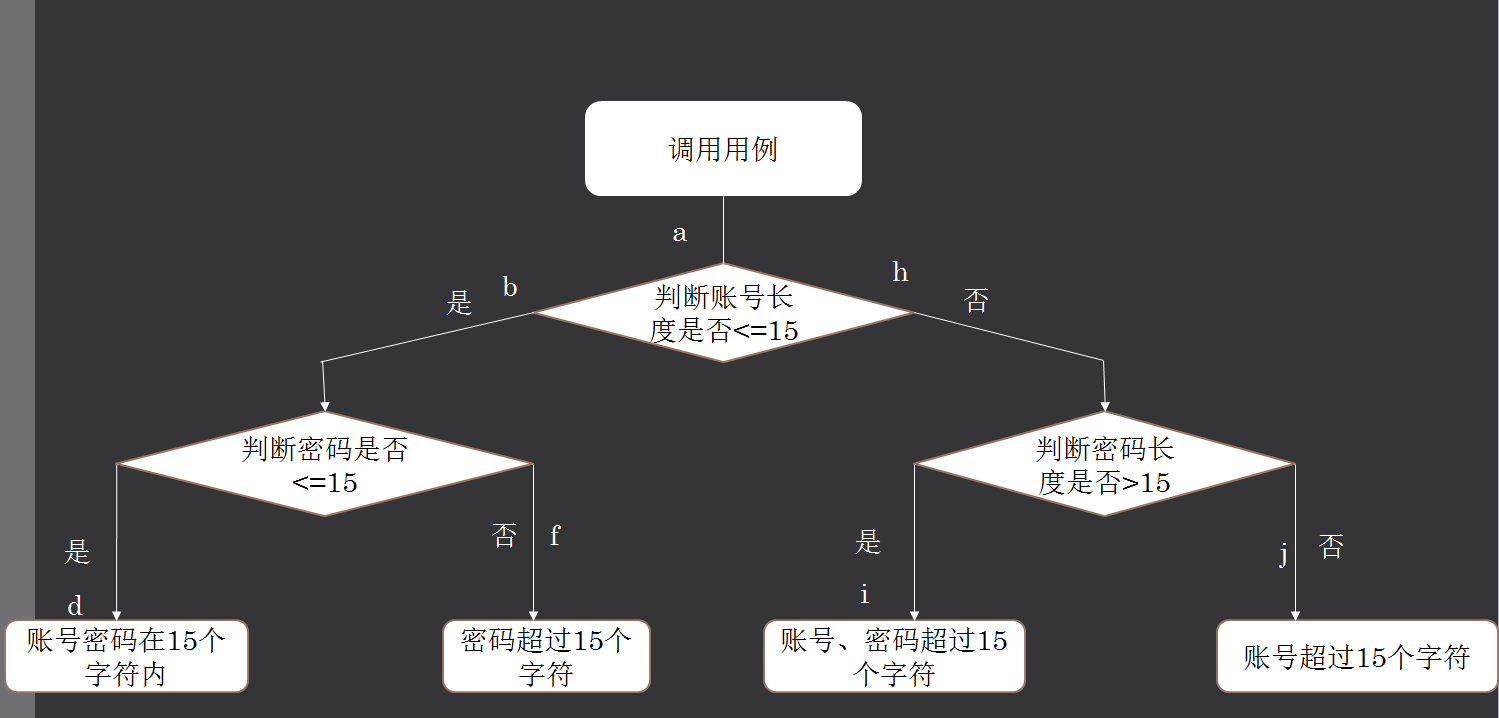
3）条件覆盖（测试用例）

|  |  |  |
| --- | --- | --- |
| 序号 | UserId | UserPwd |
| 1 | 3150139431501394(length=16) | 3150139431501394(length=16) |
| 2 | 3150139431501394(length=16) | 315013943150139 (length=15) |
| 3 | 315013943150139 (length=15) | 3150139431501394(length=16) |
| 4 | **315013943150139 (length=15)** | **315013943150139(length=15)** |

|  |  |
| --- | --- |
| 序号 | 预期输出 |
| 1 | 账号、密码超过15个字符 |
| 2 | 账号超过15个字符 |
| 3 | 密码超过15个字符 |
| 4 | 账号、密码在15个字符之内 |



4）路径覆盖



public void testDoPost3() throws ServletException, IOException{

EasyMock.expect(Request.getSession()).andReturn(UserId);

EasyMock.expect(Request.getSession()).andReturn(UserPwd);

int len = EasyMock.expect(Request.getSession()).andReturn(UserId).toString().length();

int len1 = EasyMock.expect(Request.getSession()).andReturn(UserPwd).toString().length();

EasyMock.replay();

Assert.assertFalse(servlet.isRegister(Request));

EasyMock.verify();

if(len<=15){

if(len1>15)System.out.println("账号、密码在15个字符之内");

else{

System.out.println("账号超过15个字符");

}

}

else{

if(len1<=15)System.out.println("账号、密码超过15个字符");

else{

System.out.println("密码超过15个字符");

}

}

}

|  |  |
| --- | --- |
| 测试用例 | 通过路径 |
| UserId:3150139431501394  UserPwd:3150139431501394 | ahi |
| UserId:3150139431501394  UserPwd:315013943150139 | ahj |
| UserId:315013943150139  UserPwd:3150139431501394 | abf |
| UserId:315013943150139  UserPwd:315013943150139 | abd |

#### 6.2集成测试

#### 6.3系统测试

1）恢复测试

恢复测试主要检查系统的容错能力。当系统出错时，能否在指定时间间隔内修正错误并重新启动系统。恢复测试首先要采用各种办法强迫系统失败，然后验证系统是否能尽快恢复。对于自动恢复需验证重新初始化（initialization）、检查点(check pointing mechanisms)、数据恢复(data recovery)和重新启动 (restart)等机制的正确性；对于人工干预的恢复系统，还需估测平均修复时间，确定其是否在可接受的范围内。

在测试中，我们先运行系统，然后再网页运行的时候人为关闭服务器。以此来判断系统遇到突发状况宕机后的应对情况。实际效果还可以，因为用户在出现状况之前的操作都已经被数据库记录，而且系统重新开始运行的时间间隔也没有超过十秒钟。

2）安全测试

安全测试检查系统对非法侵入的防范能力。安全测试期间，测试人员假扮非法入侵者，采用各种办法试图突破防线。例如，①想方设法截取或破译口令；②专门定做软件破坏系统的保护机制；③故意导致系统失败，企图趁恢复之机非法进入；④试图通过浏览非保密数据，推导所需信息，等等。理论上讲，只要有足够的时间和资源，没有不可进入的系统。因此系统安全设计的准则是，使非法侵入的代价超过被保护信息的价值。此时非法侵入者已无利可图。

我们的系统安全性较差，管理员的账号和密码都是统一的。如果有任何一位管理员泄露了账号和密码，那整个系统都会变得不安全。

1. 强度测试

强度测试检查程序对异常情况的抵抗能力。强度测试总是迫使系统在异常的资源配置下运行。例如，①当中断的正常频率为每秒一至两个时，运行每秒产生十个中断的测试用例；②定量地增长数据输入率，检查输入子功能的反映能力；③运行需要最大存储空间（或其他资源）的测试用例；④运行可能导致虚存操作系统崩溃或磁盘数据剧烈抖动的测试用例，等等。

在测试中，我们将每个帖子里的内容都填写到规定内的最大长度，然后在帖子的下面不断发布最长限度的评论。以此尽可能地模拟高强度的使用情况。事实中系统运行正常，只是随着帖子的长度增加，发布时等待的时间有极微小的延长。

1. 性能测试

软件部分即使满足功能要求，也未必能够满足性能要求，虽然从单元测试起，每一测试步骤都包含性能测试，只有当系统真正集成之后，在真实环境中才能全面、可靠地测试运行性能系统性能测试是为了完成这一任务。性能测试有时与强度测试相结合，经常需要其他软硬件的配套支持。

由于我们的平台并没有上线供所有用户使用，所以并不能模拟出真实的并发情况。只能在自己的电脑上不断添加各种记录。过程中，系统一直正常运行，或许是我们的测试量还不够。这项测试得到了真实环境才能有效进行。