## 6.1 测试的目的与目标

　　在此系统进行初步实现之后，开始进行对系统进行测试，找出系统中存在的Bug，通过测试，用提交的Bug报告来为以后软件的改进提供标准和参考，能够在以后的系统改进中找到依据。

测试后的软件各模块基本功能可以顺利进行，尽可能的提高软件的健壮性。

## 6.2 测试方法

* 从是否关心软件内部结构和具体实现的角度划分：黑盒测试和白盒测试；
* 从是否执行程序的角度：静态测试和动态测试；
* 从软件开发的过程按阶段划分有：单元测试、集成测试、确认测试、系统测试、验收测试、回归测试、Alpha测试、Beta测试；

单元测试又称模块测试，是针对软件设计的最小单位 ─ 程序模块（这里所说的程序模块在Java中一个模块就是一个方法），进行正确性检验的测试工作。其目的在于发现各模块内部可能存在的各种差错。单元测试需要从程序的内部结构出发设计测试用例。多个模块可以平行地独立进行单元测试。

集成测试 (组装测试、联合测试），通常在单元测试的基础上，需要将所有模块按照设计要求组装成为系统。这时需要考虑的问题是：

* 在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失；
* 一个模块的功能是否会对另一个模块的功能产生不利的影响；
* 各个子功能组合起来，能否达到预期要求的父功能；
* 全局数据结构是否有问题；
* 单个模块的误差累积起来，是否会放大，从而达到不能接受的程度。

确认测试（Validation Testing），确认测试又称有效性测试。任务是验证软件的功能和性能及其它特性是否与用户的要求一致。对软件的功能和性能要求在软件需求规格说明书中已经明确规定。它包含的信息就是软件确认测试的基础。

系统测试（System Testing），是将通过确认测试的软件，作为整个基于计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其它系统元素结合在一起，在实际运行环境下，对计算机系统进行一系列的组装测试和确认测试。系统测试的目的在于通过与系统的需求定义作比较， 发现软件与系统的定义不符合或与之矛盾的地方。

验收测试（Acceptance Testing），在通过了系统的有效性测试及软件配置审查之后，就应开始系统的验收测试。验收测试是以用户为主的测试。软件开发人员和QA（质量保证）人员也应参加。由用户参加设计测试用例，使用生产中的实际数据进行测试。在测试过程中，除了考虑软件的功能和性能外，还应对软件的可移植性、兼容性、可维护性、错误的恢复功能等进行确认。

## 6.3 测试用例

由于功能模块较多，测试内容篇幅较长，所以在本论文中只介绍登入系统的测试用例，表6.1是本系统会员登入的测试表，从测试的结果来看与期望结果完全相同。

登录系统测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能特性 | 用户登录验证 | | | | |
| 测试目的 | 验证是否输入合法的信息 | | | | |
| 测试数据 | 用户名称：1111 密码：1111 | | | | |
| 测试内容 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 输入用户姓名，按“登陆”按钮。 | 用户姓名：1111，  密码为空 | 显示警告信息“用户名或密码误!” | 显示警告信息“用户名或密码误!” | 与期望结果相同 |
| 2 | 输入密码，按“登陆”按钮。 | 用户姓名为空，密码：1111 | 显示警告信息“用户名或密码误!” | 显示警告信息“用户名或密码误!” | 与期望结果相同 |
| 3 | 输入用户姓名和密码，按“登陆”按钮。 | 用户姓名：1，  密 码：1 | 显示警告信息“用户名或密码误!” | 显示警告信息“用户名或密码误” | 与期望结果相同 |
| 4 | 输入用户姓名和密码，按“登陆”按钮。 | 用户名：1111，密 码：1111 | 正确登入到会员操作界面 | 正确登入到会员操作界面 | 与期望结果相同 |

发布功能

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能特性 | 发布失物或招领信息 | | | | |
| 测试目的 | 验证是否能正常发布失物或招领信息 | | | | |
| 测试数据 | 失物信息一则、招领信息一则 | | | | |
| 测试内容 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 登陆后，首页点击“捡到东西了”，不填写任何信息，发表 | 无 | 出现错误提示 | 提示“物品名为必填项” | 与期望结果相同 |
| 2 | 点击“捡到东西”，填写符合规定的数据，发表 | 无 | 出现发布成功提示 | 无提示，但发布成功 | 缺少提示 |
| 3 | 点击“丢东西了”，不填写任何信息，发表 | 无 | 出现错误提示 | 提示“物品名为必填项” | 与期望结果相同 |
| 4 | 点击“丢东西了”，填写符合规定的数据，发表 | 无 | 出现发布成功提示 | 无提示，但发布成功 | 缺少提示 |

搜索功能

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能特性 | 搜索网站信息 | | | | |
| 测试目的 | 验证是否能进行信息的搜索，以及结果是否正确 | | | | |
| 测试数据 | 钱包、饭 | | | | |
| 测试内容 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 搜索框输入“钱包”，点击搜索 | “钱包” | 出现带“钱包”的数据，“钱包”特殊标注 | 出现乱码 | 影响功能的错误 |
| 2 | 搜索框输入“饭”，点击搜索 | “饭” | 出现带“饭”的数据，“饭”特殊标注 | 出现乱码 | 影响功能的错误 |

在线聊天室测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能特性 | 多用户在线聊天 | | | | |
| 测试目的 | 验证是否能正常实现在线聊天功能 | | | | |
| 测试数据 | 无 | | | | |
| 测试内容 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 点击“在线交流”按钮 | 无 | 打开多人在线聊天室 | 打开多人在线聊天室 | 与期望结果相同 |
| 2 | 在文字输入框中输入文字 | 大家好，我来了！ | 文字输入框显示相应文字 | 显示“大家好，我来了！” | 与期望结果相同 |
| 3 | 输入文字后，点击“发送”按钮 | 无 | 在聊天界面显示之前输入的文字 | 在聊天界面显示之前输入的文字 | 与期望结果相同 |
| 4 | 点击“发送”按钮旁边的三角形，修改发送快捷键 | 无 | 出现快捷键修改选项 | 无响应 | 功能缺失 |

个人中心测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能特性 | 修改和查看个人信息 | | | | |
| 测试目的 | 验证功能是否正常 | | | | |
| 测试数据 | 无 | | | | |
| 测试内容 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 登陆后，点击页面上用户名字的按钮 | 无 | 进入个人信息页面 | 进入个人信息页面 | 与期望结果相同 |
| 2 | 修改个人参数界面，修改参数 | 电话：13344445555  邮箱:  aaa@aaa.com | 提示修改成功 | 提示修改成功 | 与期望结果相同 |
| 3 | 修改密码 | 密码：  12345678 | 提示修改成功，并要求重新登陆 | 提示修改成功，仍为登录状态 | 存在逻辑问题 |
| 4 | 进入“帖子信息” | 无 | 显示所有发布的贴子信息 | 显示所有发布的贴子信息 | 与期望相同 |
| 5 | 查看登录日志 | 无 | 显示所有的登录信息 | 显示所有的登录信息 | 与期望相同 |

后台管理测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能特性 | 管理后台 | | | | |
| 测试目的 | 测试管理员是否能进行后台管理操作 | | | | |
| 测试数据 | 账号：admin 密码：admin | | | | |
| 测试内容 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 点击后台登陆，输入账户密码，点击登录 | 账号：admin  密码：admin | 进入后台管理界面 | 显示登陆成功，进入管理界面 | 与预期相同 |
| 2 | 测试用户管理界面功能 | 无 | 所有功能都能实现 | 所有功能实现 | 与预期相同 |
| 3 | 测试物品管理界面功能 | 无 | 所有功能都能实现 | 所有功能实现 | 与预期相同 |
| 4 | 测试权限管理界面功能 | 无 | 所有功能都能实现 | 所有功能实现 | 与预期相同 |
| 5 | 测试菜单管理界面功能 | 无 | 所有功能都能实现 | 所有功能实现 | 与预期相同 |
| 6 | 测试系统设置界面功能 | 无 | 所有功能都能实现 | 所有功能实现 | 与预期相同 |

## 6.4 测试结论

把开始的代码写得越好，它出现的错误也就越少，你也就越能相信所做过的测试是彻底的。系统化测试以一种有序方式设法探测潜在的麻烦位置。同样，毛病最可能出现在边界，这可以通过手工的或者程序的方式检查。自动进行测试是最理想的，用得越多越好，因为机器不会犯错误、不会疲劳、不会用臆想某此实际无法工作的东西能行来欺骗自己。回归测试检查一个程序是否能产生与它们过去相同的输出。在做了小改变之后就测试是一种好技术，　能帮助我们将出现问题的范围局部化，因为新问题一般就出现在新代码里面。

测试和排错常常被说成是一个阶段，实际上它们根本不是同一件事。简单地说，排错是在你已经知道程序有问题时要做的事情。而测试则是在你在认为程序能工作的情况下，排错是在你已经知道程序有问题时要做的事情。而测试则是在你在认为程序能工作的情况下，为设法打败它而进行的一整套确定的系统化的试验。

Edsger Dijkstra有一个非常有名的说法：测试能够说明程序中有错误，但却不能说明其中没有错误。他的希望是，程序可以通过某种构造过程正确地做出来，这样就不再会有错误了，因此测试也就不必要了。这确实是个美好生活的目标，但是，对今天的实际程序而言，这仍然还只是一个理想。所以应该集中精力讨论如何测试，如何才能够更快地发现程序错误，如何才可以使得工作更有成效、效率更高。