|  |  |
| --- | --- |
| 功能测试用例库 | |
| 1.输入验证（input） | |
| 输入验证主要包括：数字输入验证、非法字符输入验证、输入长度验证、必填项验证和信息提示 | 1.数字输入验证：分别输入数字（正数、负数、零值、单精度、双精度）、字符串、空白值、空值、临界数值。不合法的输入，系统给出必要的判断提示信息 |
| 2.字符输入验证：分别输入单字节字符、双字节字符、大小写字符、特殊字符、空白值、空值。不合法的输入，系统给出必要的判断提示信息 |
| 3.日期、时间输入验证：分别输入任意字符、任意数字、非日期格式的数据、非正确日期（错误的闰年日期）、空值、空白值。不合法的输入，系统给出必要的判断提示信息。注：有些系统会不让输入当日以后或者以前的日期、时间；有些系统会通过Javascript来自动填写日期时间，这时需要注意是否能够人工主观填写输入 |
| 4.多列表选择框：测试是否能够多选，列表框中的数据是否能够显示完全。当列表框的数据过多时，需要对数据有一定格式的排序 |
| 5.单列表下拉框：测试是否能够手工输入，下拉框中的数据是否能够显示完整。当下拉框的数据很多时，需要对数据有一定的排序。如果下拉数据数值过多时，下拉框可能会超出屏幕显示范围，此种情况不能够被接收 |
| 6.大文本输入框：虽然它能够满足大数据量的输入，但最好能够显式地标明输入字符的长度限制，并且应该结合“字符输入验证”进行。需要注意的是，应该允许标点的存在 |
| 7.文件输入框输入验证：该输入框主要用做文件上传操作。在测试过程中，应该注意输入文件的扩展名。从测试角度来看，要求开发人员必须对扩展名进行输入限制，并且在适当的地方给出输入格式提示。当输入是空值等不合法的输入时，系统给出必要的判断提示信息。另外，对于上传的文件大小应该限制 |
| 8.验证码验证：做验证码输入验证时，先结合“字符输入验证”进行测试。然后注意的地方是，当界面回退或者刷新时，显示的验证码应该和实际系统验证码一致。如果验证码以图片形式显示，但图片由于其他原因（如网络）不能看到或者显示不完整，系统应该允许重新获取，最好不要做整个页面刷新 |
| 9.输入字符长度验证：输入字符的长度是否超过实际系统接收字符长度的能力。当输入超出长度时，系统给出必要的判断信息 |
| 10.必填项验证：输入不允许为空的时候，系统需要有所提示 |
| 11.格式、规则输入验证：当输入需要一定的格式时，系统需要有提示用户输入信息功能。比如身份证号码可以输入18位或者15位，部分身份证最后一位为字母X，身份证上生日与身份证号码有一定规则 |
| 12．系统错误定位的输入验证：当输入存在问题时，被系统捕获到，此时页面上的光标能够定位到发生错误的输入框 |
| 13.单选框、多选框的输入验证：单选框需要依次验证单选框的值是否有效；多选框需要依次验证多选框的值是否有效 |
| 2.操作验证（operation） | |
| 该用例库主要针对页面操作 | 1.页面链接检查：每一个链接是否都有对应的页面，并且页面之间切换正确 |
| 2.相关性检查：删除/增加一项会不会对其他项产生影响，如果产生影响，这些影响是否都正确 |
| 3.检查按钮的功能是否正确：如增、删、改、查等功能是否正确 |
| 4.重复提交表单：一条已经成功提交的记录，界面回退后再提交，看看系统是否做了处理 |
| 5.多次界面回退：检查多次使用回退的情况，在有回退的地方，回退到原来页面，再回退，重复多次，看是否出错 |
| 6.快捷键检查：（只针对PC）如：Ctrl+C、Ctrl+V等，对一些不输入信息的字段，如选人、选日期对快捷方式是否也做了限制 |
| 7.回车键检查：（只针对PC）在输入结束后直接按回车键，看系统处理如何，能否报错 |
| 8.上传下载文件检查：上传下载文件的功能是否实现，上传文件能否打开，对上传文件的格式有何规定，系统是否有解释信息，并检查系统是否能够做到 |
| 9.其他验证：在页面上图片的大小不宜太大。需要第三方软件支持时，应该给出必要的信息。比如需要jre的支持，但用户还没有安装jre，那么此时在页面上应该有显著的标识来提醒用户进行安装 |
| 3.用户注册测试用例（register） | |
| 该用例库主要针对用户注册模块 | 1.注册名输入：进行 “输入验证”。当系统有必要的输入限制提示信息时，注册用户名要按照提示信息规则进行。当系统中已经存在相同的用户时应该限制其输入，并给出适当的提示信息。一般用户注册名，输入为必填项，并且输入不能为空和空值，能够用任意字符作为用户名 |
| 2.密码输入：进行“输入验证”。一般注册系统会给出两个密码框，只有当两个输入密码框一致的时候，才能进行注册操作，否则系统应该给出必要的提示信息。密码输入应该给出一定限制，首先，不要低于6位字母（这样才可相对保证用户密码的安全性，可作为功能测试点进行），不允许有空密码的输入。其次，所有密码到数据库中后，应该是以密文形式出现，不应该是明文（这点可以通过数据库查询得到验证） |
| 3.可填项：进行“输入验证” |
| 4.登录模块测试用例（login） | |
| 该用例库主要针对用户登录模块 | 1.登录名输入：进行“输入验证”。需要注意登录名是否区分大小写和空格 |
| 2.密码输入：进行“输入验证” |
| 3.提交操作：结合“访问控制验证”。当输入正确的登录用户名和密码后，该用户能够进入到指定的操作页面；当输入的登录名和密码有误时，系统限制其登录，并且给出适当的提示信息；当遇到错误时，应该进行“错误页面测试”。 |
| 4.重设操作：当进行重设操作时，当前页面上所有输入项被清空 |
| 5.注销模块测试用例（loginout） | |
| 该用例库主要针对注销模块 | 进行注销操作：系统能够返回到首页或者一个指定的页面，确认用户session已经被注销。确认方法可以用登录后任意一个页面URL测试，当session被注销后，不能使用这个url |
| 6.增加操作测试用例（add） | |
| 该用例库主要针对增加操作 | 1.添加输入内容：进行“输入验证” |
| 2.应该限制重复增加：具体操作，利用网络传输以及服务器的延迟，多次单击“增加”按钮，经常在数据库中发现重复提交的数据 |
| 3.当增加成功或者失败后，应该有必要的信息提示 |
| 4.文件数据的增加：有些增加包含了数据库数据的增加，和一些文件的增加，此时的数据会保存在两个地方，所以测试时，需要对相关的数据做全面的验证 |
| 5.文件数据验证：进行“输入验证”之“文件输入框输入验证”。注意，当上传的文件为中文文件名时，上传到服务器后，可能会出现乱码现象。现在一般的做法是将原文件名替换成字母和数字的组合，以克服汉字文件名的弊端，另外，可以增加文件的安全性 |
| 7.删除操作测试用例（delete） | |
| 该用例库主要针对删除操作 | 1.选择需要删除的数据字段。有时候系统会根据ID来删除，有时候系统会根据名称来删除。测试的时候应该多加注意，一般要求按照ID来删除，因为可能有重名问题 |
| 2.应该限制重复删除。具体操作：利用网络以及服务器的延迟，多次单击“删除”按钮，经常在数据库中发现重复提交的数据 |
| 3.当删除的数据还有文件时，需要去验证存在数据库中的数据，以及硬盘下的文件是否都被同时删除 |
| 4.当数据被删除成功或者失败后，要有相应的信息提示 |
| 5.进行“操作验证”（operation） |
| 8.修改操作测试用例（update） | |
| 该用例库主要针对修改操作 | 1.打开需要修改的数据页面，注意与增加页面相比，只能修改部分数值，例如关键字是不能被修改的，并且二者数据用该是一致的 |
| 2.增加页面上的输入限制与修改页面的输入限制应该一致 |
| 3.修改成功或者失败后，用该有相应的信息提示 |
| 9.查询操作测试用例（query） | |
| 该用例库主要针对查询操作 | 1.条件输入查询，先进性条件输入框的“输入验证” |
| 2.条件组合查询，将多个条件进行组合查询，结果可以通过数据库验证。需要注意的事，整个数据查询和条件查询数据结果条数要一致。另外，如果遇到某天的查询时间段，有的数据库认为一天不包括零点零分，有的数据库认为包括 |
| 3.所有查询结果，必须进行一定顺序的排列。可以按照ID或者按照名称来排列 |
| 4.当查询成功或者失败后，系统应给出必要的提示信息 |
| 10.翻页操作测试用例（page turning） | |
| 该用例库主要针对翻页操作 | 1.当数据很大的时候，需要进行分页显示，每页显示的行数最好不要超过20行，每页列表上最好有序号标识，行与行之间颜色要有一定区分，这样有利于用户的查找 |
| 2.翻页按钮应该包括：首页、前一页、后一页、尾页、当前X页、工X页，这些常用按钮和显示，并且按钮都能正常翻页 |
| 3.翻页按钮的每页显示的数据要准确，确保每页查不出来的数据。最好的做法就是和数据库结合起来验证 |
| 4.页面太多，翻页数字不能全部显示时，系统应该有完善的应对机制，比如只显示当前的前三页和后三页的页码数 |
| 5.当翻到某页时，系统应该有明显的标识，标出该页面所处的页码 |
| 11.错误页面测试（error） | |
| 错误页面是在遇到系统异常的情况下产生的友好界面 | 1.当系统遇到致命错误时，不能将服务器的测试信息出现在页面上，因为这样做会带来不安全，应该给出一个合适的提示信息 |
| 2.由于系统繁忙，无法及时给出正确信息时，系统可以给出友好的错误界面，如：“请用户稍后再试”等提示信息 |

|  |  |  |
| --- | --- | --- |
| 界面测试用例库 | | |
| 1.一致性验证 | | |
| 该用例对整体风格、窗体、按钮、输入框等界面元素进行验证 | 1.整体风格一致性验证 | 查看平台中所有的用户界面，所有的界面风格是否统一，内容显示是否是正确、美观、友好，无乱、错页、变形、错别字等 |
| 2.窗体一致性验证 | 查看每个页面窗体显示是否一致，有无混乱或显示不全、不统一的页面出现。例如，菜单、大小、位置、状态和中心图片的大小是否都符合标准。注意：测试过程中需要调整浏览器的大小，进行不同像素的验证 |
| 3.按钮一致性验证 | 检查按钮图片和链接，按钮图片大小风格是否一致，点击按钮是否有效 |
| 4.输入框一致性验证 | 检查输入框大小风格，输入框高度是否一致，点击按钮是否有效 |
| 5.访问方法使用验证（只针对PC） | 检查访问方法（Tab键、鼠标移动和快捷键）的使用 |
| 6.页面链接验证 | 每一个链接是否都有对应的页面，并且页面之间切换正确 |
| 2.图形验证（graph） | | |
| 验证流程图形、图标 | 1.流程图形元素验证 | 查看流程图形是否可以正确显示，没有歧义 |
| 2.图标验证 | 查看图标显示是否正确，没有歧义并且友好显示 |
| 3.文字验证（character） | | |
| 该用例库进行文字使用、字体字号、颜色等方面的验证 | 1.文字使用验证 | 查看文字使用是否恰当，有误歧义，有误错别字 |
| 2.字体字号验证 | 查看各个页面中的字体字号显示是否一致 |
| 3.文字颜色验证 | 查看文字使用颜色是否一致 |
| 4.提示文字验证 | 查看中文版的提示文字是否为中文，英文版的提示文字是否为英文 |
| 4.浏览器兼容验证（只针对PC） | | |
| 对浏览器、脚本的兼容性进行验证 | 1.浏览器兼容性验证 | 分别查看在IE5.5、IE6.0、IE6.0+SP1、MyIE2、FirFox等常用浏览器上功能运行是否正常，界面显示是正常，风格是否一致 |
| 2.script脚本验证 | 由于不同浏览器对script脚本支持不同，所以需要在不同浏览器上进行script脚本验证 |
| 5.操作验证 | | |
| 功能说明：  根据测试需求选择测试用例 | 1.界面元素的状态验证 | 查看界面元素的状态是否正确(如有效、无效、选中等状态) |
| 2.键盘操作验证（只针对PC） | 检查界面元素是否支持键盘操作 |
| 3.鼠标操作验证（只针对PC） | 检查界面元素是否支持鼠标操作 |
| 4.缺省焦点验证 | 检查对话框中的缺省焦点是否正确 |
| 5.数据项回显验证 | 检查数据项是否能正确回显 |
| 6.常用功能验证 | 对于常用的功能，用户能否不必阅读手册就能使用 |
| 7.风险操作提示验证 | 执行有风险的操做时，检查是否有“确认”、“放弃”等提示 |
| 8.操作顺序验证 | 检查操作顺序是否合理 |
| 9.按钮排列验证 | 检查按钮排列是否合理 |
| 10.导航帮助验证 | 检查导航帮助是否明确 |
| 11.提示信息规范验证 | 检查提示信息是否规范 |