1. 功能测试
2. 链接测试（可用工具进行测试？）
3. 避免死链接情况，执行完相应操作应有返回按钮，返回到相应页面
4. 检查链接的可用性，所有链接都可以点
5. 检查链接的正确性，所有链接都跳转到预期的页面
6. 保证Web应用系统上没有孤立的页面，即只有知道正确的url才能访问
7. 链接无法访问的异常情况是否有处理
8. 表单添加
9. 快捷键检查
10. 是否支持Tab键
11. 是否支持回车键提交数据
12. 是否支持ESC键
13. 在表单填写过程中刷新，查看是否有提示，输入的信息是否丢失
14. 主键检查
15. 重复添加相同值的主键字段是否有错误提示
16. 改变已存在的主键字段值大小写进行添加是否能添加
17. 输入数据类型
18. 要求只能输入的数字不能输入其他字符串，反之亦然
19. 禁止直接输入特殊字符时，是否可以使用“粘贴”“拷贝”功能输入
20. 输入特殊字符串如null、NULL、javascript、<script>、</script>、<title>、<html>、<td>等。
21. 输入脚本函数：<script>alter("abc")</script>、document.write("abc")、<b>hello</b>
22. 特殊符号是否可正常存储，输入空格或‘“~!@#$%^&\*()\_+-={}[]|\:;”’<>,./?;”等可能导致系统错误的字符
23. 数据校验
24. 必填项不输入/输入空格，鼠标移至框外是否有错误提示
25. 输入错误格式的信息，鼠标移至框外是否有提示
26. 更改错误信息，鼠标移至框外，错误提示是否消失
27. 非必填项不输入，是否有错误提示，检查能否提交
28. 必填项放空，点击下一页或提交，页面顶部是否有提示，且不能跳转到下一页
29. 输入错误格式的信息，直接点击下一页或提交，页面顶部是否有提示
30. 后台校验，输入错误的银行卡号（或特殊字段），点击下一步是否有错误提示
31. 更改错误信息，是否可进行下一步操作
32. 输入长度检查
33. 输入字符长度为【最大值】，提交成功
34. 输入字符长度为【最小值】+1，超长不可输入
35. 输入字符长度为【最小值】，提交成功
36. 输入字符长度为【最小值】-1，提示字符长度不符合要求
37. 输入字符长度在【最小值，最大值】之间，提交成功
38. 所有字段输入字符长度为【最大值】，查看前后台数据是否走位
39. 空格检查
40. 输入字符间有空格的字符串
41. 字符前有空格
42. 字符后有空格
43. 字符前后有空格
44. 数据正确性
45. 保存后，数据是否真正插入到数据库中，注意保存后的数据正确性
46. 新增后要注意相应查询页面数据的及时更新
47. 新增信息时，多页面标签切换，保存成功后，数据是否与一致
48. 新增信息时，信息是已删除过的旧信息，是否能成功添加和修改
49. 异常处理
50. 提交后，点击BACK后再一次提交，是否正常
51. 表单提交后的提示信息，是否有做页面异常处理，如Webconnect down无法提交等
52. 提交数据时连续多次点击，检查系统会不会连续新增多条一样的数据或报错
53. 若提示不能保存，也要查看数据库里是否多了一条数据。

NOTE：表单修改和注册基本同表单添加，感觉重复了，就没再写

1. 表单删除
2. 删除某条信息时，应该有确认删除提示信息
3. 当只有一条数据时，删除后页面提示是否友好（如，没有数据）
4. 删除一条数据后，能否再添加相同的数据
5. 删除数据时，要注意相应页面数据的及时更新
6. 确定数据库是否已经删除数据
7. 登陆测试
8. 用户名/密码错误
9. 错误的用户名和密码是否能登录
10. 错误的用户名和正确的密码
11. 正确的用户名和错误的密码
12. 正确的用户名和正确的密码
13. 改变用户名和密码大小写进行登录是否能登录
14. 用户名/密码为空
15. 空的用户名和密码是否能登录
16. 空的用户名和正确的密码
17. 正确的用户名和空的密码
18. 用户名/密码组成
19. 用户名和密码是否能包含特殊符号登录
20. 用户名和密码输入超长值，超长不可输入
21. 在用户名或者密码前插入空格
22. 在用户名或者密码后插入空格
23. 在用户名或者密码中间插入空格
24. 安全性检查
25. 用户名或密码错误时没有具体说明哪个输错
26. 密码不能复制出来
27. 是否设置密码最小长度
28. 登录页面和数据库是否加密，使用足够强度的加密算法
29. 登录成功后页面url是否暴露了传输给数据库的用户名和密码
30. 用户登录是否密码输入错误次数限制或者验证码机制
31. 密码框是否有"autocomplete=off"
32. 防止SQL注入，用户名为空，密码为“1’OR ‘1=1’”进行登录，不能登录
33. 需要登录的页面直接访问url是否跳转到登录页面
34. 用户名和密码组成是否有限制（注意特殊字符：“’”“;”“[”“--””||”“xp\_”等）
35. 按OWASP Top 10 测试
36. 快捷键检查
37. Tab键能否按照顺序切换焦点
38. Enter键能否焦点于登陆按钮上
39. 搜索测试
40. 功能实现
41. 输入为空或者空格进行搜索，是否有搜索结果
42. 如果支持模糊查询，搜索名称中任意一个字符是否能搜索到数据
43. 是否支持不区分大小写搜索
44. 比较长的名称是否能查得到
45. 输入查询结果为空的查询条件，页面显示是否友好
46. 输入有查询结果的数据，逐个测试每个搜索条件的有效性
47. 更换查询条件，查看结果列表是否有刷新
48. 注意检查页面数据和数据库数据的一致性
49. 每一次新的搜索执行，都应该去除分页，显示第一页、并回到进入页面时的默认排序方式
50. 搜索条件，在分页显示时，需始终保持有效
51. 分别进行翻页、修改、添加、删除等操作后，检查查询条件不变，查询结果正确
52. 查询前后加空格
53. 安全性测试
54. 特殊符号，如单引号、%，查看是否有屏蔽这些敏感字符
55. 搜索页面等url中含有关键字的,输入html代码或JavaScript看是否在页面中显示或执行
56. 组合测试（若含多个查询条件）
57. 不同查询条件来回选择是否出现页面错误
58. 测试多个查询条件时，要注意查询条件的组合测试，可能不同组合的测试会报错，无需所有组合都测试，主要是挑用得较多的条件进行组合查询
59. 若查询条件包含时间，查询条件相应的应注意开始时间等于结束时间的情况
60. 列表测试
61. 功能实现
62. 列表数据较长排版是否合理
63. 列表的列名与列的内容正确对应
64. 没有数据时，列表是否合理显示，翻页控件是否正常
65. 多页数据时，进行中间某页，此时修改查询条件。数据和翻页控件信息就相应进行更新
66. 确认列表的默认排序方式，字段+升降续；
67. 含link的列，验证其有效性，即，点击后的跳转是否正确；
68. 分页
69. “第2页/共8页 每页 10条/共 79条”中的分页数据必须正确
70. 第一页、 上一页、下一页、最后一页的link在当前上下文有意义时显示，否则隐藏或显示为文本标签；
71. 页面能跳转到上一页/下一页/首页/尾页
72. 填入某个数字，点击“跳转到”按钮，到正确的页数；
73. 填入某个不存在的页数，点击“跳转到”按钮，跳转到首页/尾页
74. Cookie测试

测试点：Cookies的加密、自动登录、失效时间、多用户的情况、更改密码等

1. Cookie能否正常工作
2. Cookies是否起作用
3. 是否按预定的时间进行保存
4. 刷新对Cookies有什么影响
5. 是否阻止用户编辑 Cookie
6. 是否检查过如果用户浏览网站期间删除Cookie，会发生什么
7. 是否检查过如果用户关闭网站后删除Cookie，会发生什么
8. Cookie 储存的路径是否正确
9. Cookie 的信息是否正确有效，用户是否可以使用该信息连接网站
10. Cookie安全性检查
11. HttpOnly属性的设置：把Cookie的HttpOnly属性设置为True有助于缓解跨站点脚本威胁
12. Cookie过期日期设置的合理性：检查是否把Cookie的过期日期设置得过长。
13. Secure属性的设置：把Cookie的Secure属性设置为True，在传输Cookie时使用SSL连接，能保护数据在传输过程中不被篡改。
14. Cookie 信息是否经过加密
15. 文件上传
16. 文件大小类型限制
17. 上传文件是否有格式限制,是否可以上传exe文件
18. 上传文件类型错误，大小不合适的文件，是否给出了合理提示
19. 上传文件类型正确，大小不合适的文件，是否给出了合理提示
20. 上传文件类型错误，大小合适的文件，是否给出了合理提示
21. 上传文件类型正确和大小合适的文件，上传成功
22. 能否上传空文件，即0字节大小的文件
23. 通过修改扩展名的方式是否可以绕过格式限制，是否可以通过压包方式绕过格式限制
24. 上传文件
25. 上传大文件，等待过程中是否有良好提示
26. 上传重复文件，是否有覆盖提示
27. 上传正在使用中的文件，查看是否可上传成功
28. 连续多次选择不同文件，检查系统是否上传最后选择的文件
29. 是否支持多文件上传
30. 上传文件后，检查是否与源文件一致，包含目录设置等
31. 文件名称
32. 文件名称重名是否能上传
33. 文件名称含有特殊符号是否正常上传
34. 文件名称长度比较长
35. 文件名称的最大值、最小值、特殊字符（包含空格）、使用程序语句是否会对其造成影响
36. 文件下载
37. 检查用户是否有下载权限
38. 检查下载的文件的正确性
39. 右键另存为是否可以正确下载文件，并且记录下载次数
40. 单击下载是提示下载还是在页面打开
41. 若有打开选项，检查是否能直接打开，内容是否能够显示正常，页面美观性
42. 打开进行编辑，保存后，再次打开该文件，检查是否修改了文件内容
43. 下载大文件，等待过程中是否有良好提示
44. 下载重复文件至同一目录，是否有覆盖提示
45. 选择多文件时，是否能正常下载
46. 用户界面测试
47. 导航测试
48. 导航是否直观
49. Web 系统的主要部分是否可以通过主页访问
50. Web系统是否需要站点地图、搜索引擎或其他的导航器帮助
51. 测试Web 系统的页面结构
52. 导航条、菜单、连接的风格是否一致
53. 各种提示是否准确，确保用户凭直觉就知道是否还有内容，内容在什么地方
54. 图形测试
55. 要确保图形有明确的用途，图片或动画不要胡乱地堆在一起，以免浪费传输时间。图片尺寸要尽量地小，并且要能清楚地说明某件事情。
56. 验证所有页面字体的风格是否一致。
57. 背景颜色应该与字体颜色和前景颜色相搭配。
58. 图片的大小和质量也是一个很重要的因素，一般采用JPG 或GIF 压缩
59. 内容测试

检验Web 网站提供信息的正确性、准确性和相关性

1. 整理界面测试
2. 当用户浏览[Web](http://www.cntesting.com/portal/html/testing-technique/WEB-t)应用系统时是否感到舒适
3. 操作是否友好
4. 整个Web 应用系统的设计风格是否一致
5. 兼容性测试
6. 浏览器测试

兼容性测试不只是指界面在不同操作系统或浏览器下的兼容，有些功能方面的测试，也要考虑到兼容性。

测试要点：着重于对脚本、ActiveX控件、JAVA小程序、样式表、HTML、插件以及对页面的媒体文件是否可以直接播放等的兼容性测试。

IE8、9、10，Firefox 最新版，Google Chrome最新版，Safari最新版？

1. 平台兼容性

市场上有很多不同的操作系统类型，最常见的有Windows、Unix、Macintosh、Linux等。[Web](http://www.cntesting.com/portal/html/testing-technique/WEB-t)应用系统的最终用户究竟使用哪一种操作系统，取决于用户系统的配置。这样，就可能会发生兼容性问题，同一个应用可能在某些操作系统下能正常运行，但在另外的操作系统下可能会运行失败。因此，在[Web](http://www.cntesting.com/portal/html/testing-technique/WEB-t)系统发布之前，需要在各种操作系统下对[Web](http://www.cntesting.com/portal/html/testing-technique/WEB-t)系统进行兼容性[测试](http://www.cntesting.com)。

MAC和Windows系统。

1. 分辨率测试？

普通屏：640×480、800×600、1024×768、1280×1024、1600×1200等

宽屏：1280×720、1440×900、1680×1050等