Testing Web and Desktop Applications

Elena Stefanova

06/06/2017

ITSE-1391: Special Topics in Computer Programming

Table of Contents

[Web and Desktop Application Testing Checklist: 3](#_Toc481877172)

[I. Functionality Testing: 4](#_Toc481877173)

[A. Validation testing: 4](#_Toc481877175)

[B. Links/URL Testing: 4](#_Toc481877176)

[C. Web Forms Testing: 5](#_Toc481877177)

[D. Database Testing: 5](#_Toc481877178)

[E. Cookies Testing: 5](#_Toc481877179)

[F. Testing of Error Messages: 5](#_Toc481877180)

[G. Required field and optional field validation: 6](#_Toc481877181)

[H. Client-side Testing: 6](#_Toc481877182)

[Functional Test Scenarios: 6](#_Toc481877183)

[II. Usability testing: 7](#_Toc481877184)

[A. Accessibility: 7](#_Toc481877185)

[B. Identity: 7](#_Toc481877186)

[C. Navigation: 8](#_Toc481877187)

[D. Content: 8](#_Toc481877188)

[Usability Test Scenarios: 8](#_Toc481877189)

[III. Compatibility testing. 9](#_Toc481877190)

[A. Browser Compatibility Test: 9](#_Toc481877191)

[B. OS compatibility: 10](#_Toc481877192)

[C. Mobile browsing: 10](#_Toc481877193)

[Compatility Test Scenarios: 10](#_Toc481877194)

[IV. Database Testing: 10](#_Toc481877195)

[Test Scenarios for Database Testing: 11](#_Toc481877196)

[V. Crowd Testing: 11](#_Toc481877197)

[VI. Interface Testing: 12](#_Toc481877198)

[VII. Performance Testing: 12](#_Toc481877199)

[ Web Stress Testing 12](#_Toc481877201)

[ Web Load Testing 12](#_Toc481877202)

[ Soak testing 12](#_Toc481877203)

[ Spike testing 13](#_Toc481877204)

[Test scenarios: 13](#_Toc481877205)

[VIII. Security testing: 13](#_Toc481877206)

[Test Scenarios for Security Testing: 14](#_Toc481877208)

[IX. Test Scenarios 15](#_Toc481877209)

[X. What is the difference in Desktop and Web Application Testing: 23](#_Toc481877221)

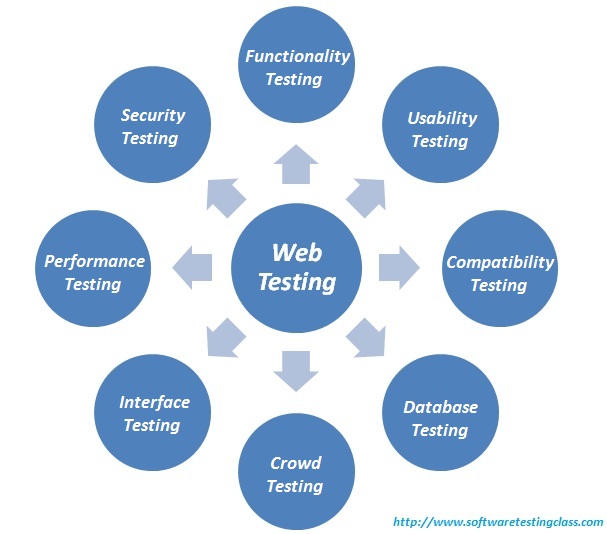
**What is Web Testing?**

**“Web testing** is the name given to Software Testing that focuses on testing the web applications.” In Web-based application is completely tested before going production environment. This could help to address the issues in web application before exposed to public like the Functional issues, web application security, web services issues, integrations issues, environment issues and its ability to handle traffic is checked. In this stage of Web Testing making effort to find out the possible bugs in the system.

**What is Desktop Testing?**

Desktop application (1) An application that runs stand alone in a desktop or laptop computer. Contrast with "Web-based application," which requires the Web browser to run. The term may be used to contrast desktop applications with mobile applications that run in smartphones and tablets.

# Web and Desktop Application Testing Checklist: (unknown)



# Functionality Testing:

**What is Functional Testing?**

* Testing the features and operational behavior of a product to ensure they correspond to its specifications.
* Testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions.

**What is the purpose or Goal of Functional testing?**

* The goal of Functional testing is to verify whether your product meets the intended functional specifications mentioned in your development documentation.

## Validation testing:

* You should make sure that if the valid HTML is used for your website. To check this you can use [W3C validator](http://validator.w3.org/).
* In functionality testing the different fields used in the website should be validated like Textboxes, dropdowns, radio options, check boxes, Combo box inputs, links etc.

Now a day’s most of the website preferred to use CSS means Cascading Style Sheet. In the market many CSS validator tools are available, one of the good tool is W3C CSS validator which will help you to validate the CSS used in your site user test.

## Links/URL Testing:

This testing is very much interesting and can be helpful for SEO of you r page. Following types of testing should be carried out for Link or URL testing:

* **Internal links**: The link which are pointing to the pages of same websites. This testing make sure that the internal links are properly linked to expected pages of your websites link like Home page, Contact Us, About Us etc.
* **External links**: The link which are pointing to the pages of external websites. This testing makes sure that the internal links are properly linked to expected pages of external websites.
* **Email links**: Such link need to make sure that the if user clicks on the email link then default email client should open with To address should be pre-filled.
* **Broken links**: Broken links are also called as Dead links. Such links are not linked to any of the pages either internal or external pages of the website. Such link generated with the spelling mistakes in the link URL or linked page is removed or no more exists. To check broken link you can use online tools to validate the broken links in the website.

## Web Forms Testing:

In *Website Testing Checklist* the web forms are the most commonly used in the websites, so it is one of the most important part of the website testing. Consider a scenario where user fills an enquiry form and click on Submit button, now what next or they just fill in the form and do nothing, the details do not get captured correctly and so are lost. While doing forms functional testing make sure that they should be consistent and should contain the required input and output controls. The data should be captured properly.

## Database Testing:

Now days with the new technologies like android and smart phones computer applications are more complex.

If the front end is more complicated then the back ends are also convoluted. As a result, introduce more complex database schema to support such intricate computer applications. So it’s more important to validate the databases to make sure the quality and website able to handle the data processing effectively.

## Cookies Testing:

A Cookie is information stored in text file on user’s hard drive (client side) by web server. This information is used later by web browser while accessing the website. Ideally the cookie is used to store the personalized user information or data in encrypted or secured manner. This is small size files which act as unique identifiers and allow websites to remember a particular user for a given time. These files are not harmful for users. Sometimes if the user’s personal information is stored in the cookie and if hackers stolen the cookie then hacker can get the confidential information which leads to security issues. That’s why the testing of Cookie is most important. There are two types of cookies Persistent Cookie and Non-Persistent Cookie.

* Test the application by disabling the Cookies
* Test the application after corrupting the cookies.
* Check the behavior of application after removing the all the cookies for the website you are testing.
* Check website writing cookies are working or not on different browser.
* Check if cookies for authenticated login are working or not.
* Check if behavior of application after deleting the cookies (sessions) by clearing cache or after cookies expired
* Check if login to application after deleting the cookies (sessions)

## Testing of Error Messages:

In the well developed website the error messages are very much helpful to guide users for success and erroneous conditions. While navigating through application if poorly designed error messages will easily misguide the end users. Many of the websites are used different interesting pages when [404 error](http://blog.hubspot.com/blog/tabid/6307/bid/33766/10-Clever-Website-Error-Messages-From-Creative-Companies.aspx) is displayed.

## Required field and optional field validation:

The proper handling of required and optional fields should be efficiently handled. Ideally the application should not be submitted unless and until all required fields are filled properly. The required error message should be displayed when user proceed with not filling the mandatory fields. It should not restrict you for proceeding further if the optional fields are not filled.

## Client-side Testing:

This type of testing is subset of Security testing. In this testing need to check if the sensitive data is not stored in the temporary internet files or stored in encrypted format like passwords, credit card information, bank number etc.

### Functional Test Scenarios:

* Test all the mandatory fields should be validated.
* Test the asterisk sign should display for all the mandatory fields.
* Test the system should not display the error message for optional fields.
* Test that leap years are validated correctly & do not cause errors/miscalculations.
* Test the numeric fields should not accept the alphabets and proper error message should display.
* Test for negative numbers if allowed for numeric fields.
* Test division by zero should be handled properly for calculations.
* Test the max length of every field to ensure the data is not truncated.
* Test the pop up message ("This field is limited to 500 characters") should display if the data reaches the maximum size of the field.
* Test that a confirmation message should display for update and delete operations.
* Test the amount values should display in currency format.
* Test all input fields for special characters.
* Test the timeout functionality.
* Test the Sorting functionality.
* Test the functionality of the buttons available
* Test the Privacy Policy & FAQ is clearly defined and should be available for users.
* Test if any functionality fails the user gets redirected to the custom error page.
* Test all the uploaded documents are opened properly.
* Test the user should be able to download the uploaded files.
* Test the email functionality of the system.
* Test the[Java](http://www.guru99.com/java-tutorial.html)script is properly working in different browsers (IE, Firefox, Chrome, safari and Opera).
* Test to see what happens if a user deletes cookies while in the site.
* Test to see what happens if a user deletes cookies after visiting a site.
* Test all the data inside combo/list box is arranged in chronological order.

# Usability testing:

**What is Usability Testing?**

* Usability testing is nothing but the User-friendliness check.
* In Usability testing, the application flow is tested so that a new user can understand the application easily.
* Basically, system navigation is checked in Usability testing.

**What is the purpose or Goal of Usability testing?**

A Usability test establishes the ease of use and effectiveness of a product using a standard Usability test practices The Usability Testing is comes under a Black Box Testing Technique where testing is to be carried out with users point of view.

The Usability testing is categories in different categories – **Accessibility, Identity, Navigation and Content**. You should keep in mid few points while testing of web application for Usability testing:

## Accessibility:

* Site Load-time should be reasonable.
* Site font size and spacing between lines should be easy to read.
* Carefully use of Flash & Add-ons in website.
* Proper ALT Tags should be used for all images present in website.
* If any internal link is broken then website should be presented with 404 error page or Not Found page.

## Identity:

* The placement of website logo should be at prominently place like right top side of the page.
* Proper tagline should be used which clearly states the purpose of the website.
* Company and Contact Information must be clearly mentioned which helps to identify the company information.

## Navigation:

* Main Navigation should be easy to find, read and use. If more than navigations are used then make sure that user should clearly understand why multiple navigations are used.
* Clear & Concise navigation labels should be used in website.
* Reasonable use of links and button in website so user will not confuse while navigating the pages.
* As a common practice the Company Logo should be linked to website Home page.
* The Site Search facility should be present on the website and make sure that the Site Search button simple and easy to access like top right of the page.

## Content:

* Headings should be clear and descriptive and SEO use of proper heading tags like H1, H2 etc.
* Make sure that the critical content should be displayed on the first screen in average screen resolution i.e. 1024×768.
* Use of consistent font styles and colors across the website help user to understand that they’re still on your site.
* Use of user friendly and meaningful keywords for URLs will help both user and search engines to understand navigation.
* Meaningful and self-explanatory titles (in the <TITLE> tag) should be used for pages. These titles are used by search engines to display in the Search result by search engines. If the improper titles are used then user will skips the your website pages link and proceed further.
* No spelling or grammatical errors mistake in content throughout the page.
* Alt text should be present on Images
* No broken images
* Your task is to validate all for UI testing
* Follow some standard on content building on web page
* All content should be legible & easy to understand.
* Dark color infuriates the users, so avoid using dark colors in the theme.
* Proper size images should be placed on web page
* All the anchor text links should be working properly.

### Usability Test Scenarios:

* Web page content should be correct without any spelling or grammatical errors
* All fonts should be same as per the requirements.
* All the text should be properly aligned.
* All the error messages should be correct without any spelling or grammatical errors and the error message should match with the field label.
* Tool tip text should be there for every field.
* All the fields should be properly aligned.
* Enough space should be provided between field labels, columns, rows, and error messages.
* All the buttons should be in a standard format and size.
* Home link should be there on every single page.
* Disabled fields should be grayed out.
* Check for broken links and images.
* Confirmation message should be displayed for any kind of update and delete operation.
* Check the site on different resolutions (640 x 480, 600x800 etc.?)
* Check the end user can run the system without frustration.
* Check the tab should work properly.
* Scroll bar should appear only if required.
* If there is an error message on submit, the information filled by the user should be there.
* Title should display on each web page
* All fields (Textbox, dropdown, radio button etc) and buttons should be accessible by keyboard shortcuts and the user should be able to perform all operations by using keyboard.
* Check if the dropdown data is not truncated due to the field size and also check whether the data is hardcoded or managed via administrator.

# Compatibility testing.

**What is Compatibility testing?**

* Compatibility testing is used to determine if your software is compatible with other elements of a system with which it should operate, e.g. Browsers, Operating Systems, or hardware.

**What is the purpose or Goal of Compatibility testing?**

* The purpose of Compatibility testing is to evaluate how well software performs in a particular **browser**(IE8, IE9, IE10, IE11, Chrome, Firefox, Safari, Opera etc), **Operating Systems**(Windows XP, Windows 7, Vista, Linux, Mac etc) **hardware or software**. The Compatibility testing is to make sure that “**Is web application show correctly across different devices**?”

## Browser Compatibility Test:

Web applications are rendering differently on different browsers. The objective of browser compatibility testing is to ensure that no any errors on the different web browsers while rendering the sites. In Browser Compatibility Testing you need to ensure that your web application is being displayed properly on different browsers. Also check AJAX, JavaScript and authentication are functioning correctly.

## OS compatibility:

In new technology newer graphics designs are used & different APIs are used which may not work on different Operating systems. Also on rendering of different objects like text fields, buttons may display different on different Operating System. So testing of web application should be carried out on different OS like Windows, MAC, Solaris, Unix, Linux with different OS flavors.

## Mobile browsing:

In latest Mobi technology you also test out Mobile Browser Compatibility too. It may be possible of Compatibility issues on Mobile browsers. So in the new Mobi technology age you testing of web pages on mobile browsers should be carried out.

### Compatility Test Scenarios:

* Test the website in different browsers (IE, Firefox, Chrome, Safari and Opera) and ensure the website is displaying properly.
* Test the HTML version being used is compatible with appropriate browser versions.
* Test the images display correctly in different browsers.
* Test the fonts are usable in different browsers.
* Test the java script code is usable in different browsers.
* Test the Animated GIF's across different browsers.

**Tool for Compatibility Testing:**

Spoon.net: Spoon.net provides access to thousands of applications (Browsers) without any installs. This tool helps you to test your application on different browsers on one single machine**.**

# Database Testing:

**What is Database Testing?**

* In Database testing backend records are tested which have been inserted through the web or desktop applications. The data which is displaying in the web application should match with the data stored in the Database.

Testing activities would include-

* Check if queries are executed without any errors.
* Creating, updating or deleting data in database should maintain the data integrity.
* More time should not take to execute the queries, if required tune the queries for better performance.
* Check load on database while executing heavier queries & check the result.
* Collect data from database & represent on the web pages correctly.

### Test Scenarios for Database Testing:

* Verify the database name: The database name should match with the specifications.
* Verify the Tables, columns, column types and defaults: All things should match with the specifications.
* Verify whether the column allows a null or not.
* Verify the Primary and foreign key of each table.
* Verify the Stored Procedure:
* Test whether the Stored procedure is installed or not.
* Verify the Stored procedure name
* Verify the parameter names, types and number of parameters.
* Test the parameters if they are required or not.
* Test the stored procedure by deleting some parameters
* Test when the output is zero, the zero records should be affected.
* Test the stored procedure by writing simple[SQL](http://www.guru99.com/sql.html)queries.
* Test whether the stored procedure returns the values
* Test the stored procedure with sample input data.
* Verify the behavior of each flag in the table.
* Verify the data gets properly saved into the database after the each page submission.
* Verify the data if the DML (Update, delete and insert) operations are performed.
* Check the length of every field: The field length in the back end and front end must be same.
* Verify the database names of QA, UAT and production. The names should be unique.
* Verify the encrypted data in the database.
* Verify the database size. Also test the response time of each query executed.
* Verify the data displayed on the front end and make sure it is same in the back end.
* Verify the data validity by inserting the invalid data in the database.
* Verify the Triggers.

# Crowd Testing:

Crowd testing is when a large group of perfect strangers try your product then give you phenomenally helpful feedback on usability, bugs and features.

To test the software application Crowd testing can be used. It not limited to web applications, but for all kinds of applications including mobile application testing. Crowdtesting is dependent on the quality of the crowd. Also it depends on a crowd that is composed out of a large group of diver’s people. *It used do system tests for performance and usability testing. Simply this*is complementary to ‘normal’ testing. The mainly complicated job of crowd testing is determining a good enough crowd.

# Interface Testing:

In the Interface testing mainly three areas should be covered: Web Server, Application Server and Database Server. Ensure that all the communications between these all servers should be carried out correctly. Verify that if connection between any servers is reset or lost then what is happing. Check if any request interrupts in-between then how application is responding. On returns of any error from web server or database server to application server then error should be Errors are handled properly & display such errors to the user.

* Web Server: Check if all web requests are accepting and not any requests are denied or leakages.
* Application Server: Check if request is sending correctly to the any server & displayed correctly. Check if errors are catch properly & displayed to admin user.
* Database Server: Check if database server is returns correct result on query request.

Check if all three servers are connected to each & test request is processing correctly. And any error in between then error should be displayed to user.

# Performance Testing:

**What is Performance Testing?**

Performance testing is conducted to evaluate the compliance of a system or component with specified performance requirements.

* Web Stress Testing- It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.
* Web Load Testing- It is the simplest form of testing conducted to understand the behaviour of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, application server, etc. are also monitored.
* Soak testing - Soak Testing also known as endurance testing, is performed to determine the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues. The main aim is to discover the system's performance under sustained use.
* Spike testing - Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the work load.

**This would include:**

* Check if response times of Website application under different speeds of connections
* Check if site handles many simultaneous user requests at same time.
* Check if how your web application sustain under the peak loads
* Check if large input data from users.
* Check the behavior of web application if simultaneous connection to Database.
* Check if how the web site pulls through if crash occurs due to peak load.
* Check if optimization methods such as reduce load times by enabling cache on browser client and server side, gzip compression etc
* Check if any hardware memory leakage errors

### Test scenarios:

* To determine the performance, stability and scalability of an application under different load conditions.
* To determine if the current architecture can support the application at peak user levels.
* To determine which configuration sizing provides the best performance level.
* To identify application and infrastructure bottlenecks.
* To determine if the new version of the software adversely had an impact on response time.
* To evaluate product and/or hardware to determine if it can handle projected load volumes.

Hence to overcome the above problems we should use Performance testing tool. Below is the list of **some popular testing tools**.

* Apache JMeter
* Load Runner
* Borland Silk Performer.
* Rational Performance Tester
* WAPT
* NEO LOA

# Security testing:

**What is Security Testing?**

1. Security Testing involves the test to identify any flaws and gaps from a security point of view. Some of the major aspects of web security testing are:

* **Penetration Testing**
* **Password cracking**
* **Vulnerability**
* **URL manipulation**
* **SQL injection**
* **Network Scanning**
* **Log Review**
* **Integrity Checkers**
* **Virus Detection**

Testing Activities will include-

* Check if unauthorized access to secure pages, if user changes from “https” to “http” (secure to non-secure) in secure pages then proper message should be display and vice versa.
* Check if accessing internal pages directly entering URLs in browser. If login is required then user should redirected to login page or appropriate message should be displayed.
* Most of the information related to transactions, error messages, login attempts should be logged in log file.
* Check if restricted files are able to access for download.
* Check if internal Web directories or files are not accessible unless & until not configured for download.
* Check if CAPTCHA is added & working properly for logins to prevents automates logins attempts.
* Check if try to access others information by changing parameter in query string. For example if you are editing the information & in URL you are seeing UserID = 123, try to change this parameter values & check if application is not providing the other users information. It should display Access denied for this user to view others users information.
* Check if sessions are got expired after pre-defined amount of time if user not using session.
* Check if user not able to pass login page for invalid username/password combination.
* Check if user is navigated to encrypted SSL pages for secure website.

### Test Scenarios for Security Testing:

1. Verify the web page which contains important data like password, credit card numbers, secret answers for security question etc should be submitted via HTTPS (SSL).
2. Verify the important information like password, credit card numbers etc should display in encrypted format.
3. Verify password rules are implemented on all authentication pages like Registration, forgot password, change password.
4. Verify if the password is changed the user should not be able to login with the old password.
5. Verify the error messages should not display any important information.
6. Verify if the user is logged out from the system or user session was expired, the user should not be able to navigate the site.
7. Verify to access the secured and non secured web pages directly without login.
8. Verify the “View Source code” option is disabled and should not be visible to the user.
9. Verify the user account gets locked out if the user is entering the wrong password several times.
10. Verify the cookies should not store passwords.
11. Verify if, any functionality is not working, the system should not display any application, server, or database information. Instead, it should display the custom error page.
12. Verify the[SQL](http://www.guru99.com/sql.html)injection attacks.
13. Verify the user roles and their rights. For Example The requestor should not be able to access the admin page.
14. Verify the important operations are written in log files, and that information should be traceable.
15. Verify the session values are in an encrypted format in the address bar.
16. Verify the cookie information is stored in encrypted format.
17. Verify the application for Brute Force Attacks

# Test Scenarios (unknown)

**General Test Scenarios**

1. All mandatory fields should be validated and indicated by asterisk (\*) symbol  
2. Validation error messages should be displayed properly at correct position  
3. All error messages should be displayed in same CSS style (e.g. using red color)  
4. General confirmation messages should be displayed using CSS style other than error messages style (e.g. using green color)  
5. Tool tips text should be meaningful  
6. Dropdown fields should have first entry as blank or text like ‘Select’  
7. Delete functionality for any record on page should ask for confirmation  
8. Select/deselect all records options should be provided if page supports record add/delete/update functionality  
9. Amount values should be displayed with correct currency symbols  
10. Default page sorting should be provided  
11. Reset button functionality should set default values for all fields  
12. All numeric values should be formatted properly  
13. Input fields should be checked for max field value. Input values greater than specified max limit should not be accepted or stored in database  
14. Check all input fields for special characters  
15. Field labels should be standard e.g. field accepting user’s first name should be labeled properly as ‘First Name’  
16. Check page sorting functionality after add/edit/delete operations on any record  
17. Check for timeout functionality. Timeout values should be configurable. Check application behavior after operation timeout  
18. Check cookies used in an application  
19. Check if downloadable files are pointing to correct file paths  
20. All resource keys should be configurable in config files or database instead of hard coding  
21. Standard conventions should be followed throughout for naming resource keys  
22. Validate markup for all web pages (validate HTML and CSS for syntax errors) to make sure it is compliant with the standards  
23. Application crash or unavailable pages should be redirected to error page  
24. Check text on all pages for spelling and grammatical errors  
25. Check numeric input fields with character input values. Proper validation message should appear  
26. Check for negative numbers if allowed for numeric fields  
27. Check amount fields with decimal number values  
28. Check functionality of buttons available on all pages  
29. User should not be able to submit page twice by pressing submit button in quick succession.  
30. Divide by zero errors should be handled for any calculations  
31. Input data with first and last position blank should be handled correctly

[**GUI**](http://www.softwaretestinghelp.com/gui-testing-on-smart-devices-%e2%80%93-testing-guidelines/)**Test Scenarios**

1. All fields on page (e.g. text box, radio options, dropdown lists) should be aligned properly  
2. Numeric values should be right justified unless specified otherwise  
3. Enough space should be provided between field labels, columns, rows, error messages etc.  
4. Scroll bar should be enabled only when necessary  
5. Font size, style and color for headline, description text, labels, infield data, and grid info should be standard as specified in SRS  
6. Description text box should be multi-line  
7. Disabled fields should be grayed out and user should not be able to set focus on these fields  
8. Upon click of any input text field, mouse arrow pointer should get changed to cursor  
9. User should not be able to type in drop down select lists  
10. Information filled by users should remain intact when there is error message on page submit. User should be able to submit the form again by correcting the errors  
11. Check if proper field labels are used in error messages  
12. Dropdown field values should be displayed in defined sort order  
13. Tab and Shift+Tab order should work properly  
14. Default radio options should be pre-selected on page load  
15. Field specific and page level help messages should be available  
16. Check if correct fields are highlighted in case of errors  
17. Check if dropdown list options are readable and not truncated due to field size limit  
18. All buttons on page should be accessible by keyboard shortcuts and user should be able to perform all operations using keyboard  
19. Check all pages for broken images  
20. Check all pages for broken links  
21. All pages should have title  
22. Confirmation messages should be displayed before performing any update or delete operation  
23. Hour glass should be displayed when application is busy  
24. Page text should be left justified  
25. User should be able to select only one radio option and any combination for check boxes.

**Test Scenarios for Filter Criteria**

1. User should be able to filter results using all parameters on the page  
2. Refine search functionality should load search page with all user selected search parameters  
3. When there is at least one filter criteria is required to perform search operation, make sure proper error message is displayed when user submits the page without selecting any filter criteria.  
4. When at least one filter criteria selection is not compulsory user should be able to submit page and default search criteria should get used to query results  
5. Proper validation messages should be displayed for invalid values for filter criteria

**Test Scenarios for Result Grid**

1. Page loading symbol should be displayed when it’s taking more than default time to load the result page  
2. Check if all search parameters are used to fetch data shown on result grid  
3. Total number of results should be displayed on result grid  
4. Search criteria used for searching should be displayed on result grid  
5. Result grid values should be sorted by default column.  
6. Sorted columns should be displayed with sorting icon  
7. Result grids should include all specified columns with correct values  
8. Ascending and descending sorting functionality should work for columns supported with data sorting  
9. Result grids should be displayed with proper column and row spacing  
10. Pagination should be enabled when there are more results than the default result count per page  
11. Check for Next, Previous, First and Last page pagination functionality  
12. Duplicate records should not be displayed in result grid  
13. Check if all columns are visible and horizontal scroll bar is enabled if necessary  
14. Check data for dynamic columns (columns whose values are calculated dynamically based on the other column values)  
15. For result grids showing reports check ‘Totals’ row and verify total for every column  
16. For result grids showing reports check ‘Totals’ row data when pagination is enabled and user navigates to next page  
17. Check if proper symbols are used for displaying column values e.g. % symbol should be displayed for percentage calculation  
18. Check result grid data if date range is enabled

**Test Scenarios for a Window**

1. Check if default window size is correct  
2. Check if child window size is correct  
3. Check if there is any field on page with default focus (in general, the focus should be set on first input field of the screen)  
4. Check if child windows are getting closed on closing parent/opener window  
5. If child window is opened, user should not be able to use or update any field on background or parent window  
6. Check window minimize, maximize and close functionality  
7. Check if window is re-sizable  
8. Check scroll bar functionality for parent and child windows  
9. Check cancel button functionality for child window

[**Database Testing**](http://www.softwaretestinghelp.com/database-testing-%e2%80%93-practical-tips-and-insight-on-how-to-test-database/)**Test Scenarios**

1. Check if correct data is getting saved in database upon successful page submit  
2. Check values for columns which are not accepting null values  
3. Check for data integrity. Data should be stored in single or multiple tables based on design  
4. Index names should be given as per the standards e.g. IND\_<Tablename>\_<ColumnName>  
5. Tables should have primary key column  
6. Table columns should have description information available (except for audit columns like created date, created by etc.)  
7. For every database add/update operation log should be added  
8. Required table indexes should be created  
9. Check if data is committed to database only when the operation is successfully completed  
10. Data should be rolled back in case of failed transactions  
11. Database name should be given as per the application type i.e. test, UAT, sandbox, live (though this is not a standard it is helpful for database maintenance)  
12. Database logical names should be given according to database name (again this is not standard but helpful for DB maintenance)  
13. Stored procedures should not be named with prefix “sp\_”  
14. Check is values for table audit columns (like createddate, createdby, updatedate, updatedby, isdeleted, deleteddate, deletedby etc.) are populated properly  
15. Check if input data is not truncated while saving. Field length shown to user on page and in database schema should be same  
16. Check numeric fields with minimum, maximum, and float values  
17. Check numeric fields with negative values (for both acceptance and non-acceptance)  
18. Check if radio button and dropdown list options are saved correctly in database  
19. Check if database fields are designed with correct data type and data length  
20. Check if all table constraints like Primary key, Foreign key etc. are implemented correctly  
21. Test stored procedures and triggers with sample input data  
22. Input field leading and trailing spaces should be truncated before committing data to database  
23. Null values should not be allowed for Primary key column

**Test Scenarios for Image Upload Functionality**

*(Also applicable for other file upload functionality)*  
1. Check for uploaded image path  
2. Check image upload and change functionality  
3. Check image upload functionality with image files of different extensions (e.g. JPEG, PNG, BMP etc.)  
4. Check image upload functionality with images having space or any other allowed special character in file name  
5. Check duplicate name image upload  
6. Check image upload with image size greater than the max allowed size. Proper error message should be displayed.  
7. Check image upload functionality with file types other than images (e.g. txt, doc, pdf, exe etc.). Proper error message should be displayed  
8. Check if images of specified height and width (if defined) are accepted otherwise rejected  
9. Image upload progress bar should appear for large size images  
10. Check if cancel button functionality is working in between upload process  
11. Check if file selection dialog shows only supported files listed  
12. Check multiple images upload functionality  
13. Check image quality after upload. Image quality should not be changed after upload  
14. Check if user is able to use/view the uploaded images

**Test Scenarios for Sending Emails**

*(Test cases for composing or validating emails are not included)*  
*(Make sure to use dummy email addresses before executing email related tests)*  
1. Email template should use standard CSS for all emails  
2. Email addresses should be validated before sending emails  
3. Special characters in email body template should be handled properly  
4. Language specific characters (e.g. Russian, Chinese or German language characters) should be handled properly in email body template  
5. Email subject should not be blank  
6. Placeholder fields used in email template should be replaced with actual values e.g. {Firstname} {Lastname} should be replaced with individuals first and last name properly for all recipients  
7. If reports with dynamic values are included in email body, report data should be calculated correctly  
8. Email sender name should not be blank  
9. Emails should be checked in different email clients like Outlook, Gmail, Hotmail, Yahoo! mail etc.  
10. Check send email functionality using TO, CC and BCC fields  
11. Check plain text emails  
12. Check HTML format emails  
13. Check email header and footer for company logo, privacy policy and other links  
14. Check emails with attachments  
15. Check send email functionality to single, multiple or distribution list recipients  
16. Check if reply to email address is correct  
17. Check sending high volume of emails

**Test Scenarios for Excel Export Functionality**

1. File should get exported in proper file extension  
2. File name for the exported Excel file should be as per the standards e.g. if file name is using timestamp, it should get replaced properly with actual timestamp at the time of exporting the file  
3. Check for date format if exported Excel file contains date columns  
4. Check number formatting for numeric or currency values. Formatting should be same as shown on page  
5. Exported file should have columns with proper column names  
6. Default page sorting should be carried in exported file as well  
7. Excel file data should be formatted properly with header and footer text, date, page numbers etc. values for all pages  
8. Check if data displayed on page and exported Excel file is same  
9. Check export functionality when pagination is enabled  
10. Check if export button is showing proper icon according to exported file type e.g. Excel file icon for xls files  
11. Check export functionality for files with very large size  
12. Check export functionality for pages containing special characters. Check if these special characters are exported properly in Excel file

**Performance Testing Test Scenarios**

1. Check if page load time is within acceptable range  
2. Check page load on slow connections  
3. Check response time for any action under light, normal, moderate and heavy load conditions  
4. Check performance of database stored procedures and triggers  
5. Check database query execution time  
6. Check for load testing of application  
7. Check for stress testing of application  
8. Check CPU and memory usage under peak load condition

[**Security Testing**](http://www.softwaretestinghelp.com/security-testing-of-web-applications/)**Test Scenarios**

1. Check for SQL injection attacks  
2. Secure pages should use HTTPS protocol  
3. Page crash should not reveal application or server info. Error page should be displayed for this  
4. Escape special characters in input  
5. Error messages should not reveal any sensitive information  
6. All credentials should be transferred over an encrypted channel  
7. Test password security and password policy enforcement  
8. Check application logout functionality  
9. Check for Brute Force Attacks  
10. Cookie information should be stored in encrypted format only  
11. Check session cookie duration and session termination after timeout or logout  
11. Session tokens should be transmitted over secured channel  
13. Password should not be stored in cookies  
14. Test for Denial of Service attacks  
15. Test for memory leakage  
16. Test unauthorized application access by manipulating variable values in browser address bar  
17. Test file extension handing so that exe files are not uploaded and executed on server  
18. Sensitive fields like passwords and credit card information should not have auto complete enabled  
19. File upload functionality should use file type restrictions and also anti-virus for scanning uploaded files  
20. Check if directory listing is prohibited  
21. Password and other sensitive fields should be masked while typing  
22. Check if forgot password functionality is secured with features like temporary password expiry after specified hours and security question is asked before changing or requesting new password  
23. Verify CAPTCHA functionality  
24. Check if important events are logged in log files  
25. Check if access privileges are implemented correctly

# What is the difference in Desktop and Web Application Testing:

|  |  |
| --- | --- |
| **Desktop Application Testing** | **Web Application Testing** |
| 1. Application which run on single system /computer or workstation. | 1. It also runs on two more computers. |
| 2. There is no server or client and it is a standalone application. | 2.There are two or more systems in which one is server and other is client. The application is loaded on server and there is not executable file. |
| 3. It has a single user. | 3. It has unlimited users. |
| 4. There is no client and server. | 4. Here we may or may not have any knowledge about the server location. |
| 5.  It is done on a single machine or work station. | 5. It is performed on 3 tier application generally. |
| 6. In Desktop applications we test application features like GUI, backend and load. | 6. In Web application testing we test the application functionality, OS compatibility and browser compatibility. |
| 7. Here the environment is the user machine. | 7. Here the environment is web browsers. |
| 8. These are desktop driven application. | 8. Web Testing is URL driven testing. |
| 9. in Desktop Application there is only one user accessing it and the application may or may not require authentic access. | 9. In Web Application there are u unlimited users and it can be accessed by all the users. |

**Example of Desktop and Web Application Testing:**

**Desktop Application**: Applications like MS Excel, MS Word, and Outlook. Some desktop applications made by technologies like HTML and JS which allow the developers to write code. Thus the desktop applications are also made of these technologies.

**Web Application**: These applications 3 –tier usually developed in Internet. These have a browser, a web server and a database. These applications are generally built in HTML, Javascript, XML etc and the web server is generally built in Java, ASP, JavaScript, VBScript, PHP. The Database servers would be oracle, sql server, mysql etc.

# Bibliography

unknown. (n.d.). Retrieved from http://www.softwaretestinghelp.com

unknown. (n.d.). Retrieved from http://www.softwaretestinghelp.com/

unknown. (n.d.). Retrieved from http://www.softwaretestinghelp.com/