Web Testing: Complete Guide To Your Web Application Testing

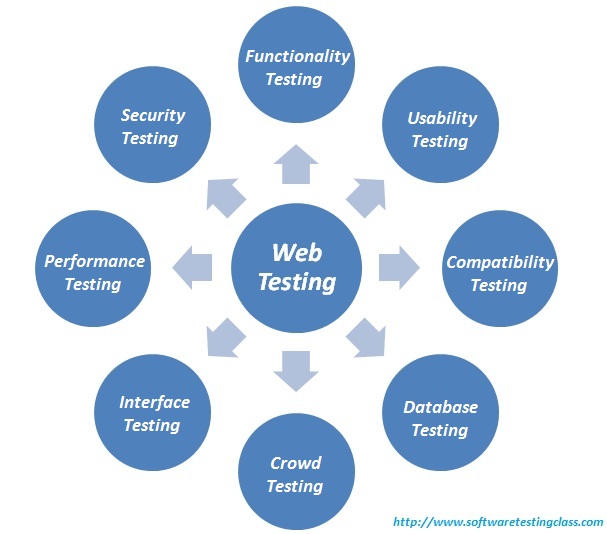
Prior to see more details on web application testing, first let’s see the definition of the Web Testing.

**What is Web Testing?**

**“Web testing** is the name given to [Software Testing](http://www.softwaretestingclass.com/) 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.

**Web Application Testing Checklist:**

Let see what all testing is to be carried out on in ***software web testing***. The testing is totally based on your web testing requirements but following is the standard checklist of web testing:

[](http://www.softwaretestingclass.com/wp-content/uploads/2012/09/web-testing.jpg)

**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.

Let’s start with the Web Application Testing Checklist:

**[1] Functionality Testing:**

The following testing must be carried out while doing Website [Functionality testing](http://www.softwaretestingclass.com/functional-testing/):

**A) 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.

**B) 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.

**C) 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.

**D) 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.

**E) 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)

**F) 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.

**G) 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.

**H) 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.

**2. 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](http://www.softwaretestingclass.com/category/black-box-testing-techniques/) 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:

**A) Accessibility: (Add Bullets)**

* 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.

**B) 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.

**C) 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.

**D) 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.

**3. 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**?”

This would include

»Check on different browsers and its versions.

» Check on different Operating systems and its versions.

» Check on different hardware configurations

» Check on different network environments.

» Check on different screen resolutions

**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**.**

**4. 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.

**To perform the Database testing, the tester should be aware of the below mentioned points**:

* The tester should understand the functional requirements, business logic, application flow and database design thoroughly.
* The tester should figure out the tables, triggers, store procedures, views and cursors used for the application.
* The tester should understand the logic of the triggers, store procedures, views and cursors created.
* The tester should figure out the tables which get affected when insert update and delete (DML) operations are performed through the web or desktop applications.

**With the help of the above mentioned points, the tester can easily write the test scenarios for Database testing.**

**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.

**5. 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.

**6. 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.

**7. 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

**General 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.

**How to do Performance testing? By Manual Testing or by Automation**

Practically it is not possible to do the performance testing manually because of some drawbacks like:

* More number of resources will be required.
* Simultaneous actions are not possible.
* Proper system monitoring is not available.
* Not easy to perform the repetitive task.

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

**8. 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

[**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



Though this is a common checklist, I recommend preparing a standard testing checklist tailored to your specific needs using below test cases in addition with application specific tests.

**Importance of Using Checklist for Testing:**  
– Maintaining a standard repository of reusable test cases for your application will ensure the most common bugs will be caught more quickly.  
– Checklist helps to quickly complete writing test cases for new versions of the application.  
– Reusing test cases help to save money on resources to write repetitive tests.  
– Important test cases will be covered always making it almost impossible to forget.  
– Testing checklist can be referred by developers to ensure most common issues are fixed in development phase itself.

*Few notes to remember:*  
1) Execute these scenarios with different user roles e.g. admin user, guest user etc.  
2) For web applications these scenarios should be tested on [multiple browsers](http://www.softwaretestinghelp.com/best-cross-browser-testing-tools-to-ease-your-browser-compatibility-testing-efforts/) like IE, FF, Chrome, and Safari with versions approved by client.  
3) Test with different screen resolutions like 1024 x 768, 1280 x 1024, etc.  
4) Application should be tested on variety of displays like LCD, CRT, Notebooks, Tablets, and Mobile phones.  
4) Test application on different platforms like Windows, Mac, Linux operating systems.

**Comprehensive Testing Checklist for Testing Web and Desktop Applications:**

**Assumptions:** Assuming that your application supports following functionality  
– Forms with various fields  
– Child windows  
– Application interacts with database  
– Various search filter criteria and display results  
– Image upload  
– Send email functionality  
– Data export functionality

**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/)**and Usability 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

# How To Test Responsive Website - Sample Test cases and Examples!!

We all have experienced this at least once, when we are trying to surf website on mobile to see a particular area of the website. We are zooming in by taping on the screen like a thousand times. It is annoying and at the same time we lose our interest. As a result it could be a potential loss to the website business. As the organization has lost a customer by not opting responsive website for its business.

In this article, we are going to learn “How to test responsive website”. Also learn about related factors which are worth considering while testing Responsive web design.

What is exact meaning of Responsive Website?

Responsive website design is the web design approach aimed to view websites on different devices, resolution. Checking that able render and adjust the page to provide optimal viewing experience.

Let’s say if user switches from mobile view to desktop/laptop/ipad view or vice versa. In such cases website page should render automatically and set best possible screening experience.

Responsive website is compatible with multiple devices like all browsers, resolutions, screen sizes on different screen size desktops and devices like Smartphone, tablets etc. It is very important to test the responsiveness of the website for excellent user experiences.

[](http://www.softwaretestingclass.com/wp-content/uploads/2016/12/testing-responsive-design.png)

When a website is developed, main concern of the client is RWD (Responsive Web Design), since it is easier to see and shop from our mobile device then to open our PC and surf. We carry our smartphone everywhere with us so it is an extremely flexible option for us. No matter what size of the smartphone screen you use, RWD should be compatible with all of the varieties. That is why most of the “ready to use” web designed or themes come with responsiveness. So developer just needs to write the code for the website and not to worry about the responsiveness.

Components of Responsive Website:

A website designed with RWD mainly uses with fluid grid, flexible images, percentage based grids, Media queries with CSS3 styles as per following:

* To resize the page element most common concept is used in RW design is fluid grid. Instead of using absolute units like points or pixels for element, it uses relative units like percentages.
* RWD are implemented in different coding languages like PHP, .Net, Java and many new technologies which supports RWD.
* For image resizing to prevent the images goes beyond the defined grid on different resolution, flexible images are resized using relative units.
* Media queries are most commonly used to set different CSS style for different resolutions. It looks for the width of the browser to apply the appropriate size based on the width defined.
* Using power of HTML and CSS can implement RWD with ease, few of the features of CSS and HTML allows you to resize images, screen resolution automatically.
* Responsive website design works on the lower and upper limit to resize the website. Most of the time the breakpoints are defined based on the width of the website, if the website goes above or below of the break-point the appropriate web design gets applied.

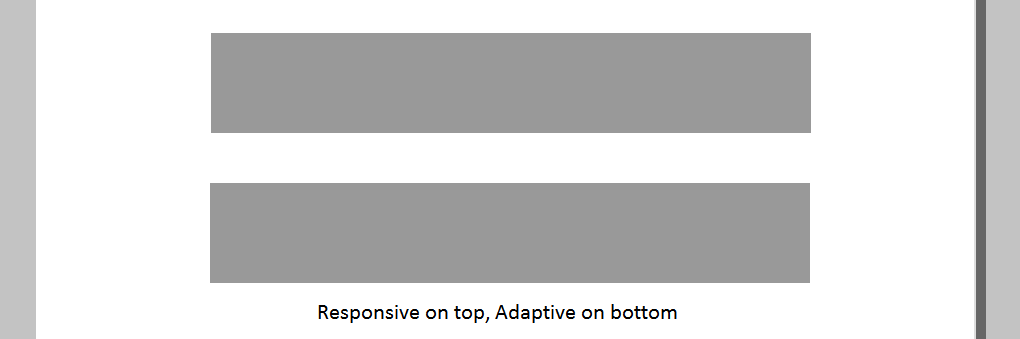
Difference Between Responsive and Adaptive Design

This is most common question is being asked while testing RWD. I think this is very good question.

The responsive sites and adaptive sites are sites which changes the appearance of based on the browser width. The main difference is responsive website changes the appearance after changes in width of browser every point. **For example**, if the browser 1024 px wide and width changes below or above 1024 px then layout will respond accordingly. Responsive sites adjust the layout fluidly not considering on which device is being used.

The adaptive site changes the appearance after changes in width of browser beyond the specific defined points. **For example**, box control with 500 pixels wide displayed on site when width of the browser is more than 800 px and if width below 800 px then box should reduce the size to 300 pixels. Adaptive design is mainly used when page need to serve different on different browser or devices.

Checkout below animation for exemplifies the exact difference:

[](http://www.softwaretestingclass.com/wp-content/uploads/2016/12/Responsive-and-Adaptive-Design.gif)

Challenges in testing responsive website:

Nowadays many people are using mobiles or tablets to test surf websites. Accessing website over mobile device is quite cumbersome if those are not web design is not responsive.

One of the most challenging part of responsive website testing is to test the websites on all browsers, resolutions, screen sizes and Operating Systems etc. However it is not realistic to test the website with all above combinations.

By and large almost every tester start testing responsiveness of website by resizing browser to check the break-point sizing of tablet, mobiles, desktop with different resolutions and window to fit the viewpoint.

This exercise to quick check the rendering issues after resizing the browser. On the other hand, testing on real mobile devices with features like landscape, portrait, zooming, finger print access etc. and similarly experience on desktop like mouse pointers, mouse clicks etc.

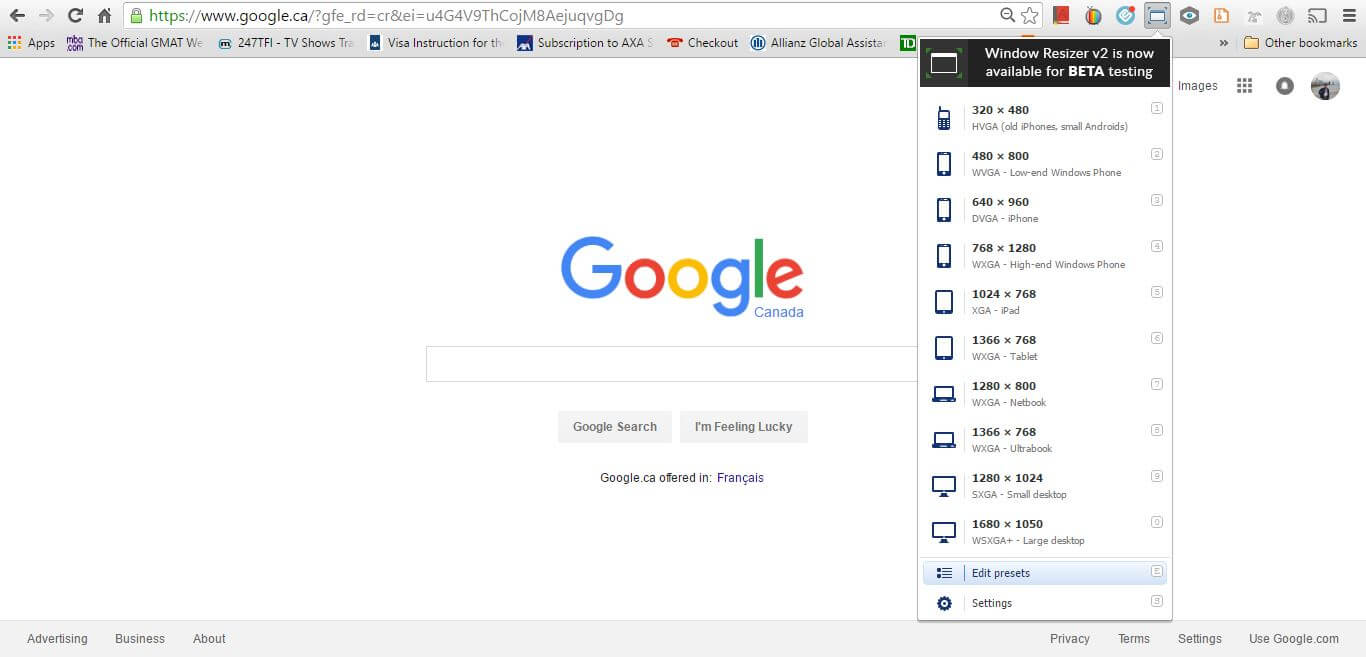
To do test pass or RWD we need to think about all these variations.

How to Test a responsive website:

The following are some of the ways to test such web design:

**1) Changing the size of your browser**: when you are testing the responsiveness of a web design, you can easily check the compatibility by changing the size of your browser. For example, if you go to below link: [google.com](http://www.google.com/) and try to reduce the size of your browser screen, you would notice that the search bar has also moved to fix in the window size. Reduce it further and make it of a size similar to a smartphone, you will notice that the web page has adjusted to the size of the viewport of the device successfully. This method is perfect for a quick visual test for responsiveness.

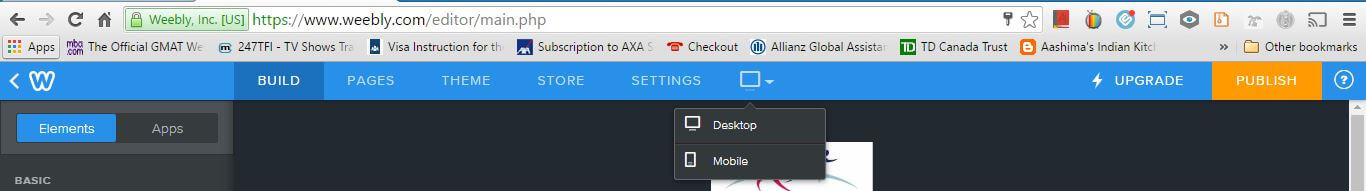
If you have used the Google chrome web browser then you may notice that at the top right hand side of the browser, there is the viewport sizing option available as shown below. By using this option you can change the size of the viewport to that of a smartphone or a tablet and notice the responsiveness of the web design for that particular website. It is a quick and very easy option available to test the responsiveness of the web design across the variable viewport sizes of the multiple devices.

[](http://www.softwaretestingclass.com/wp-content/uploads/2016/12/responsive-web-design-1.jpg)

**2) Emulators:** These are the virtual mobiles or web based simulators depicting mobile device like environment. When you test RWD, you have to test it for various mobile devices and it could be costly to buy so many devices for testing purposes, emulators in such a case solve the purpose. It shows how a website would look and function on the mobile. Therefore, without buying the actual handset, we can test the web design responsiveness on the emulators which could save us the money.

[](http://www.softwaretestingclass.com/wp-content/uploads/2016/12/testing-responsive-design-emulators.png)

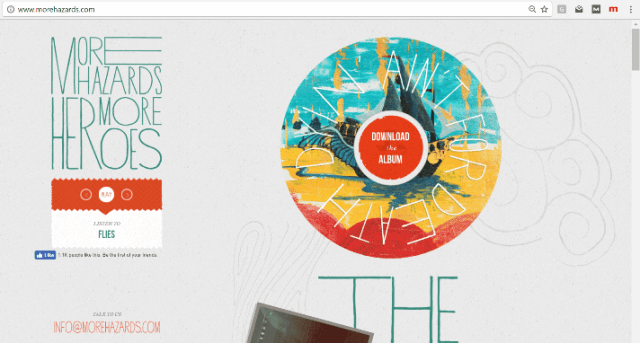
**3) DevTools:** Some website builder tools shows different options to see how your website design would look on different devices. For example, website builder by Godaddy, Weebly, wordpress, etc. are some of the development tools which have that option. I have attached a screenshot from weebly for reference. They are really very helpful since they have got 3 different sizes namely desktop, mobile, iPad or tablets.

[](http://www.softwaretestingclass.com/wp-content/uploads/2016/12/responsive-web-design-2.jpg)

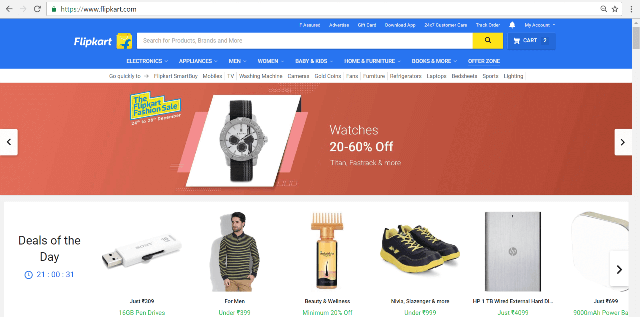
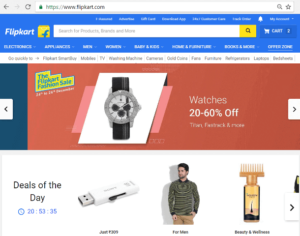
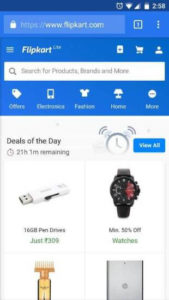
1. **Online responsive checker:** There are online free checker website available. You just need to put your website URL and check for responsiveness. They are very easy to use and you can get results in less than one minute.

Test cases for testing responsive website

Some of the important factors to keep in mind while testing the RWD are as follows.

* To verify all the images on the web page are displayed properly on all the different devices and resolution.
* To verify text and headings on the web page are properly aligned.
* To verify all the clickable links on the web page are readable and work as expected.
* To verify scrolling of the web page works as expected.
* To verify if there are input boxes and text areas to enter data then we need to make sure that the text entered is displayed properly on the web page and they are aligned as expected.
* To verify image size, Font size and font type are consistent across all the web pages.
* To verify if contents of the page are displayed consistent on all resolutions.
* To verify the color changes after hover over the elements.
* To verify the consistency of color combination on different resolutions.
* To verify images, text, different controls are not going beyond the screen border.
* To verify if there should not be any horizontal scrolling bar since everything should be fit according to the size of the screen.
* To verify on rotating your mobile device, all the contents should be rotated and displayed as expected without any technical glitch.
* To verify if the user able to click on clickable area.
* To verify padding of elements on the edges.
* To verify if enter text in input box are displayed as expected without any UI glitches.
* [](http://www.softwaretestingclass.com/wp-content/uploads/2016/12/morehazards-1.png)
* Launch the URL on medium screen like tablet:
* [](http://www.softwaretestingclass.com/wp-content/uploads/2016/12/morehazards-tablet.png)
* Launch the URL on small screen like mobile:
* [](http://www.softwaretestingclass.com/wp-content/uploads/2016/12/morehazards-mobile.png)

## Example #2: www.flipkart.com

* Launch the URL on large screen like laptop/desktop:
* [](http://www.softwaretestingclass.com/wp-content/uploads/2016/12/Flipkart-1.png)
* Launch the URL on medium screen like tablet:
* [](http://www.softwaretestingclass.com/wp-content/uploads/2016/12/Flipkart-tablet.png)
* Launch the URL on small screen like mobile:
* Checkout on loading Flipkart website on mobile showing **Filpkart Lite version**.
* [](http://www.softwaretestingclass.com/wp-content/uploads/2016/12/flipkart-mobile-lite-1.jpg)
* These were some of the key points to remember. You should know the break-point of the web design, there are some websites with huge images and cannot be shrieked after a certain extent. It is important to know the breakpoints and decide how it will be shown on a smaller device.  Sometimes, a picture tends to be hidden on the smaller device due to its huge size depending on the different screen size options as per display need. As simple is that, your website should look good and it should not break the website layouts when the viewport size is changed when that website is viewed on different devices.

## ****Conclusion:****

* In this article, we have discussed the number of ways to test RWD (Responsive Web Design). At the same time it also depends on the kind of website that you are testing and what are the permissible data that you want to show on the website which may not be appropriate to show on a small screen devices due to their viewport size limitations. Therefore, the responsive website should be the one that should give the website the best look and feel across all variable screen sized devices.

**What is the difference in Desktop, Client Server and Web Application Testing:**

|  |  |  |
| --- | --- | --- |
| **Desktop Application Testing** | **Client Server Application Testing** | **Web Application Testing** |
| 1. Application which run on single system /computer or workstation. | 1. Client Server testing runs on two more computers. | 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 an executable file is installed on the client machines. | 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 multiple user but limited number. | 3. It has unlimited users. |
| 4. There is no client and server. | 4. In this we have knowledge about the server location. | 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 2 tier application generally. | 5. It is performed on 3 tier application generally. |
| 6. In Desktop applications we test application features like GUI, backend and load. | 6. In Client Server we test features of applications like GUI on both sides, functionality. | 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 usually the intranet. | 7. Here the environment is web browsers. |
| 8. These are desktop driven application. | 8. These are menu driven application testing. | 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 Client Server application there are limited users and the application user are already known before. They might have an username/password to access the application. | 9. In Web Application there are u unlimited users and it can be accessed by all the users. |

**Example of Desktop, Client Server 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.

**Client Server Application**: These applications are 2 –tier developed in LAN usually. They have a front end in form of forms and reports. The front end would allow the user to manipulate and fetch data. These applications are developed in C#, VB, Core Java etc and would user Database like MySQL, Oracle, Sybase.

**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.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case id** | **Test cases** | **Priority** | **Preconditions** | **Input test data** | **Steps to be executed** | **Expected results** | **Actual results** | **Pass/fail** | **Comments** |
| 1 | Test if user is able to login successfully. | A | User must be registered already | correct username,correct password | 1)Enter input(correct )username and password on the respective fields 2)click submit/login | User must successfully login to the web page | (note down the results you have observed) |  |  |
| 2 | Test if unregistered users is not able to login to the site | A |  | incorrect username,incorrect password | 1)Enter input(incorrect )username and password on the respective fields 2)click submit/login | Proper error must be displayed and prompt to enter login again | (note down the results you have observed) |  |  |
| 3 | Test with valid username and empty password such that login must get failed | B | User must be registered already | valid username and empty password | 1)enter the valid username in the user id and enter no password in the password field | Proper error must be displayed and prompt to enter login again | (note down the results you have observed) |  |  |
| 4 | Test with empty username and valid password such that login must get failed | B | registered user's password | empty username and valid password | 1)leave the username empty in the user id and enter a valid user's password in the password field | Proper error must be displayed and prompt to enter login again | (note down the results you have observed) |  |  |
| 5 | Test with empty username and empty password and check if login fails | A | - | - | 1)Enter nothing in the mail id and password field 2)click submit button | Proper error must be displayed and prompt to enter login again |  |  |  |
| 6 | Check of the password is masked on the screen i.e., password must be in bullets or asterisks | B |  | some password(can be a registered/unregistered) | 1) Enter the password field with some characters | The password field should display the characters in asterisks or bullets such that the password is not visible on the screen |  |  |  |
| 7 | Check if the login function handles case sensitivity | B | registered user's password which is originally in lower case changed to upper case or vice versa | case changed username /password | 1)Enter the case changed username /password in the respective field and 2)click login button | Login must fail saying incorrect username/password |  |  |  |
| 8 | After logging in try to copy/cut the password and paste it on another screen(passwords are usually in \* such that its not visible on the screen) | B |  | Registered user's login id and password | 1)Enter username and password in the respective fields. Copy the password field's content(which is in \*s) 3)paste the content on another screen | password shouldn’t get pasted / password should not be visible on the screen |  |  |  |
| 9 | Verify account lock | B |  | Registered user's login id and incorrect password | 1)Try to login with a registered user name and incorrect password for more than 3 times | Account should be locked and access should be granted only after gettting certain assurance from the user |  |  |  |
| 10 | Check if on selecting back button (after logging out) if the user is not signed in | B |  | Registered username and password | 1)Login with registered username and password 2)once your are logged in, sign out of the site 3)now press back button | User shouldn’t be signed in to his account rather a general webpage must be visible |  |  |  |
| 11 | Verify the url without logging into to the site | B |  | Registered username and password | 1) Login to the site using registered username and password 2)copy and save the url of the logged in page 3)logout of the site 4)now paste the copied url on the browser | the url should not redirect to a logged in page but to a logged out page of the site |  |  |  |
| 12 | Automatic logout of the site when pressing backspace button | B | User must be registered already | Registered username and password | 1) Login to the site using registered username and password 2)now press backspace | User must logout of the site properly |  |  |  |

**Mobile Testing: Complete Guide to Test your Mobile Apps**

Some or all of the following[Testing](http://www.guru99.com/software-testing.html)types may be performed depending on your mobile testing requirements **[](http://cdn.guru99.com/images/Mobile_Testing.png)**

* Functional testing
* Performance testing
* Security testing
* Usability testing
* Compatibility testing
* Recoverability Testing

**Functional testing:**

The functional testing of Mobiles normally consists in the areas of testing user interactions as well as testing the transactions. The various factors which are relevant in functional testing are

1. Type of application based upon the business functionality usages (banking, gaming, social or business)
2. Target audience type (consumer, enterprise, education)
3. Distribution channel which is used to spread the application (e.g. Apple App Store, Google play, direct distribution)

The most fundamental test scenarios in the functional testing can be considered as :

1. To validate whether all the required mandatory fields are working as required.
2. To validate that the mandatory fields are displayed in the screen in a distinctive way than the non-mandatory fields.
3. To validate whether the application works as per as requirement whenever the application starts/stops.
4. To validate whether the application goes into minimized mode whenever there is an incoming phone call. In order to validate the same we need to use a second phone, to call the device.
5. To validate whether the phone is able to store, process and receive SMS whenever the app is running. In order to validate the same we need to use a second phone to send sms to the device which is being tested and where the application under test is currently running.
6. To validate that the device is able to perform required multitasking requirements whenever it is necessary to do so.
7. To validate that the application allows necessary social network options such as sharing, posting and navigation etc.
8. To validate that the application supports any payment gateway transaction such as Visa, Mastercard, Paypal etc as required by the application.
9. To validate that the page scrolling scenarios are being enabled in the application as necessary.
10. To validate that the navigation between relevant modules in the application are as per the requirement.
11. To validate that the truncation errors are absolutely to an affordable limit.
12. To validate that the user receives an appropriate error message like “Network error. Please try after some time” whenever there is any network error.
13. To validate that the installed application enables other applications to perform satisfactorily, and it does not eat into the memory of the other applications.
14. To validate that the application resumes at the last operation in case of a hard reboot or system crash.
15. To validate whether the installation of the application can be done smoothly provided the user has the necessary resources and it does not lead to any significant errors.
16. To validate that the application performs auto start facility according to the requirements.
17. To validate whether the application performs according to the requirement in all versions of Mobile that is 2g, 3g and 4g.
18. To perform regression testing to uncover new software bugs in existing areas of a system after changes have been made to them. Also rerun previously performed tests to determine that the program behavior has not changed due to the changes.
19. To validate whether the application provides an available user guide for those who are not familiar to the app

**Performance testing:**

This type of testing’s fundamental objective is to ensure that the application performs acceptably under certain performance requirements such as access by a huge number of users or the removal of a key infrastructure part like a database server.

The general test scenarios for performance testing in a Mobile application are:

1. To determine whether the application performs as per the requirement under different load conditions.
2. To determine whether the current network coverage is able to support the application at peak, average and minimum user levels.
3. To determine whether the existing client-server configuration setup provides the required optimum performance level.
4. To identify the various application and infrastructure bottlenecks which prevent the application to perform at the required acceptability levels.
5. To validate whether the response time of the application is as per as the requirements.
6. To evaluate product and/or hardware to determine if it can handle projected load volumes.
7. To evaluate whether the battery life can support the application to perform under projected load volumes.
8. To validate application performance when network is changed to WIFI from 2G/3G or vice versa.
9. To validate each of the required the CPU cycle is optimization
10. To validate that the battery consumption, memory leaks, resources like GPS, Camera performance is well within required guidelines.
11. To validate the application longevity whenever the user load is rigorous.
12. To validate the network performance while moving around with the device.
13. To validate the application performance when only intermittent phases of connectivity is required.

**Security testing:**

[](http://cdn.guru99.com/images/Mobile_Testing1.png)

The fundamental objective of security testing is to ensure that the application’s data and networking security requirements are met as per guidelines.

The following are the most crucial areas for checking the security of Mobile applications.

1. To validate that the application is able to withstand any brute force attack which is an automated process of trial and error used to guess a person’s username, password or credit-card number.
2. To validate whether an application is not permitting an attacker to access sensitive content or functionality without proper authentication.
3. To validate that the application has a strong password protection system and it does not permit an attacker to obtain, change or recover another user’s password.
4. To validate that the application does not suffer from insufficient session expiration.
5. To identify the dynamic dependencies and take measures to prevent any attacker for accessing these vulnerabilities.
6. To prevent from[SQL](http://www.guru99.com/sql.html)injection related attacks.
7. To identify and recover from any unmanaged code scenarios.
8. To ensure whether the certificates are validated, does the application implement Certificate Pinning or not.
9. To protect the application and the network from the denial of service attacks.
10. To analyze the data storage and data validation requirements.
11. To enable the session management for preventing unauthorized users to access unsolicited information.
12. To check if any cryptography code is broken and ensure that it is repaired.
13. To validate whether the business logic implementation is secured and not vulnerable to any attack from outside.
14. To analyze file system interactions, determine any vulnerability and correct these problems.
15. To validate the protocol handlers for example trying to reconfigure the default landing page for the application using a malicious iframe.
16. To protect against malicious client side injections.
17. To protect against malicious runtime injections.
18. To investigate file caching and prevent any malicious possibilities from the same.
19. To prevent from insecure data storage in the keyboard cache of the applications.
20. To investigate cookies and preventing any malicious deeds from the cookies.
21. To provide regular audits for data protection analysis.
22. Investigate custom created files and preventing any malicious deeds from the custom created files.
23. To prevent from buffer overflows and memory corruption cases.
24. To analyze different data streams and preventing any vulnerabilities from these.

**Usability testing:**

[](http://cdn.guru99.com/images/Mobile_Testing2.png)

The usability testing process of the Mobile application is performed to have a quick and easy step application with less functionality than a slow and difficult application with many features. The main objective is to ensure that we end up having an easy-to-use, intuitive and similar to industry-accepted interfaces which are widely used.

1. To ensure that the buttons should have the required size and be suitable to big fingers.
2. To ensure that the buttons are placed in the same section of the screen to avoid confusion to the end users.
3. To ensure that the icons are natural and consistent with the application.
4. To ensure that the buttons, which have the same function should also have the same color.
5. To ensure that the validation for the tapping zoom-in and zoom-out facilities should be enabled.
6. To ensure that the keyboard input can be minimized in an appropriate manner.
7. To ensure that the application provides a method for going back or undoing an action, on touching the wrong item, within an acceptable duration.
8. To ensure that the contextual menus are not overloaded because it has to be used quickly.
9. To ensure that the text is kept simple and clear to be visible to the users.
10. To ensure that the short sentences and paragraphs are readable to the end users.
11. To ensure that the font size is big enough to be readable and not too big or too small.
12. To validate the application prompts the user whenever the user starts downloading a large amount of data which may be not conducive for the application performance.
13. To validate that the closing of the application is performed from different states and verify if it re-opens in the same state.
14. To ensure that all strings are converted into appropriate languages whenever a language translation facility is available.
15. To ensure that the application items are always synchronized according to the user actions.
16. To ensure that the end user is provided with a user manual which helps the end user to understand and operate the application who may be not familiar with the application’s proceedings

Usability testing is normally performed by manual users since only human beings can understand the sensibility and comfort ability of the other users.

**Compatibility testing:**

Compatibility testing on mobile devices is performed to ensure that since mobile devices have different size, resolution, screen, version and hardware so the application should be tested across all the devices to ensure that the application works as desired.

The following are the most prominent areas for compatibility testing.

1. To validate that the user Interface of the application is as per the screen size of the device, no text/control is partially invisible or inaccessible.
2. To ensure that the text is readable for all users for the application.
3. To ensure that the call/alarm functionality is enabled whenever the application is running. The application is minimized or suspended on the event of a call and then whenever the call stops the application is resumed.

**Recoverability Testing**

1. Crash recovery and transaction interruptions
2. Validation of the effective application recovery situation post unexpected interruption/crash scenarios.
3. Verification of how the application handles a transaction during a power failure (i.e. Battery dies or a sudden manual shutdown of the device)
4. The validation of the process where the connection is suspended, the system needs to re-establish for recovering the data directly affected by the suspended connection.

**Other Important Checks:**

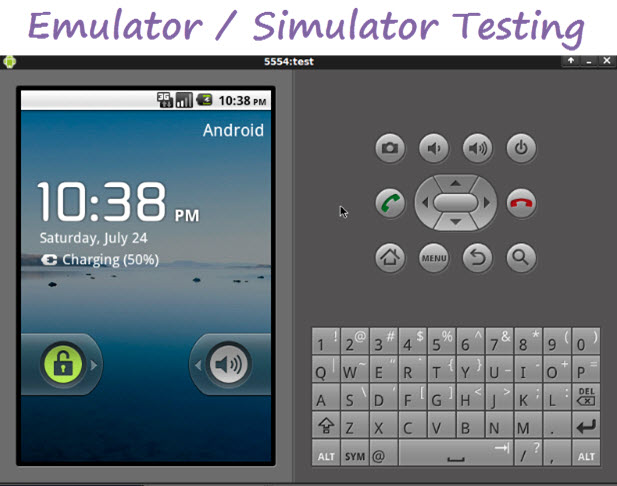
1. Installation testing (whether the application can be installed in a reasonable amount of time and with required criterion)
2. Uninstallation testing (whether the application can be uninstalled in a reasonable amount of time and with required criterion)
3. Network test cases (validation of whether the network is performing under required load or not, whether the network is able to support all the necessary applications during the testing procedures)
4. Check Unmapped keys
5. Check application splash screen
6. Continued keypad entry during interrupts and other times like network issues
7. Methods which deal with exiting the application
8. Charger effect while an application is running in the background
9. Low battery and high performance demand
10. Removal of battery while an application is being performed
11. Consumption of battery by application
12. Check Application side effects

**Real Device Vs Emulator Testing: Ultimate Showdown**

**Real Testing Device:**Testing on real device allows you to run your mobile applications and checks its functionality. Real device[Testing](http://www.guru99.com/software-testing.html)assures you that your application will work smoothly in customer handsets.

[](http://cdn.guru99.com/images/1stimage.jpg)

**Emulators:**Emulator is a software program that allows your mobile to imitate the features of another computer or mobile software you want them to imitate by installing them to your computer or Mobile.

[](http://cdn.guru99.com/images/2ndimage.jpg)

**Difference between the emulator and simulator based testing:**

Both Emulators and Simulators are virtual devices. A virtual device is not the real phone but software which gives same functionality as the real phone (except few functionality like the camera).

But there are some differences between an Emulator and Simulator describe as below –

|  |  |
| --- | --- |
| **The simulator based testing** | **The emulator based testing** |
| Simulator's objective is to simulate the internal state of an object as close as possible to the internal state of an object. | The emulator aims at emulating or mimicking as close as possible the outer behavior of an object |
| Simulators are preferable whenever the testing team needs to test the mobile's external behavior like calculating, making transactions and so forth. | Emulators are preferable whenever the testing team needs to test the mobile's internal behavior like its internal hardware, firmware and so forth. |
| Simulators are written in high level languages. | Emulators are written in machine-level assembly languages. |
| The simulators can be difficult in terms of debugging purpose. | Emulators are more suitable when it comes to debugging purpose |
| A simulator is just a partial re-implementation of the original software . | Often an emulator comes as a complete re-implementation of the original software . |

**Relative Advantages of real device-based application and emulator/simulator based testing**

|  |  |  |
| --- | --- | --- |
| Issue | Emulator Testing | Real Device Testing |
| Situation-based application | There are specific situations where the deadline to produce text execution results are short and purchasing the required mobile devices may be not possible. Thereby it might be necessary to use the emulator/simulator in these circumstances for testing the relevant mobile applications which need to be tested. | The real device allows the testers to test almost all the real time scenarios which can be tested for the mobile applications. These devices are operated using fingers and simulate real-life usage. They also help in situation Real context: is it easy to use the app on the train, or while walking down the street? The situation about in bright sunlight or in the rain? |
| Feeling of closeness towards the real handheld devices | The wide gamut of mobile devices creates the problems, whereby the testers are not confident about which mobile devices to invest in for testing, considering the budget constraints. Emulator/simulator (s) is tailor made for this type of situation(s). | The real device allows the testers to test even usability issues like the look and feel of the application, color resolution of the screen, whether the picture is bright or not under both day and night conditions and so on. |
| Ease of availability | Emulator/simulator(s) are in most cases open and free software which can be very easily downloaded from Internet and ready to be tested for. | The real devices allow stringent performance testing issues like working with a real time transport application for 15 hours continuously which cannot be successfully simulated by the emulators. |
| Ease of opening an Web application through URL | It is easier to do web application testing when it comes to opening the web application. The user just needs to copy and paste the application URL. | Testing on real devices provides more in terms of reliability. |
| Capturing screenshots of the situations where defects appear | Capturing issue of screenshots over simulator is very easy with the simulator since we just need to use Microsoft office facilities. | Testing with real world devices is very helpful in terms of interoperability testing. |
| Simulation of validation of battery scenarios | The emulator/simulators are not able to simulate the battery issues. | Real world devices can easily perform the same. |
| Validation of incoming interrupts | The emulator/simulators are not able to simulate the incoming interrupts for SMS as well as the incoming calls. | Real world devices can easily simulates incoming interrupts. |
| Validation of exact color displays | The emulator/simulator is not able to properly emulate/simulate the exact color display of the devices when the real device is in sunlight or in black. | Real world devices can easily simulates the exact color displays. |
| Validation of the performance | The performance of the emulator/simulator tends to be slower than the original devices at times. | The original devices tend to perform faster than the emulator or the simulators. |
| Simulating memory related issues | The memory available at the emulator/simulator tends to be far more than the real devices so this may create misconception for the users who would be using the same validations. | The memory storage level of the devices tend to be far less than the emulators thus it may |

**Disadvantages of Simulators and Real device**

|  |  |
| --- | --- |
| **Emulators/ Simulators** | **Real Device** |
| The emulator/simulator is not always the best type of solution for scenarios such as the ones whereby the testing team needs to validate the performance of the application for a longer period of time. | The real devices are costly compared to the emulator/simulators. Thereby projects under budget and timeline constraints may risk profitability as well as the viability of the overall project. |
| The emulator/simulator is suitable mostly for certain types of functional test case executions. | There is a very wide variety of mobile devices from apple to Samsung to android and to Symbian and so on. Considering this wide range of mobile devices it is very hard for the testing team to arrange all sorts of mobile devices while working under considerable amount of budget and timeline related constraints. |
| The emulator/simulator can some time not be supportive of certain types of application and in these cases the testing team may need to purchase software patches which may not always be free but could be costly at times. | Real Mobile devices when used in the developing stage for unit testing and similar purposes could turn out to be harder to connect to the IDE than the emulators and this causes tremendous problems for debugging and in a project with timeline constraints this may very well hamper the overall conclusion of the project. |
| Not all the emulator/simulator supports the complete gamut of mobile applications. For example the bada simulator supports the Maemo (such as Nokia N900), Symbian Touch (such as Nokia N8) and Symbian non-touch (such as Nokia E71) but it does not support other mobile devices like Android. As per as the application testing functionalities are concerned, bada does not support direct web browsing testing but it allows the user to test and create only webapps and widgets. | In order to test with the real world devices, the devices need to be always connected to the USB port of the machines. So if the USB ports are not working properly, the testing would not be possible. Without providing adequate security measures mobile devices (if they happen to be costly like the apple Iphone) may be lost or stolen thus hampering the overall effort. Increasing security may also go on to increase the overall expenditure involved with the project. |
|  | The user has to type manually the URL for opening up the web application which is needed to be tested. To solve this particular issue, the tester may need to create phone bookmarks, short URL services or sending URLs to mobile using Bluetooth connection or creating the webpage that contains some URL-s. The adoption of these procedures would ensure that a lot of very important memory space may be eaten up thus impacting on the overall performance of the application. |

**Conclusion**

Considering the significant role the mobile applications plays, nowadays, in our day to day life, testing of these applications are going to evolve and thus they require a lot of testing to make them work as required.Testing in both the simulator/emulator as well as the real world devices is necessary to maintain strong standards and quality assurance.

Careful deliberation of both the pros and cons of mobile emulators and real devices, it would be worthwhile to reach at the conclusion that the optimal mobile testing solution for enterprises is neither putting all the eggs into the basket of the real devices nor putting them into the emulator but rather what we need is **an optimum combination of both**.

Emulators can be considered as very suitable for the initial stages of application development.

However, to avoid the costly scenario of releasing a business-critical application with defects, enterprises need to ensure that they perform the major part of their mobile testing on real devices before the application goes into production.

Each organization needs to strategize and plan carefully to determine at what stage to introduce real devices, they also need to decide how many devices are sufficient to cover market needs, and what could be the best possible option to adopt for managing those devices.

Best practices would indicate that actual development should use emulators (and a few reference real handsets) in order to speed up the debugging of the application during the coding phase, while sanity, smoke testing, performance, interoperability and network feasibility and regression testing should be done on real handsets.

It is also an emerging practice to ensure that the developers use the emulator for fast execution during the development phase whereas then the testing team should test with the real device during the testing phase in order to ensure overall quality assurance goals and targets.To save on cost, they can consider using Virtual Mobile Testing tools. These services offer developer to test their application on wide variety of handsets using different mobile networks geographically located throughout the world (useful for applications using GPS). Such services are offered on hourly basis and are very cost effective compared to buying new phones.

**Mobile App Performance Testing: Strategy, CheckList, Tools**

For any mobile app, performance is very critical. If your Mobile App does not perform well, the end user will uninstall your app find another application that performs better.

Your Mobile application needs to be tested thoroughly before releasing it to end user.

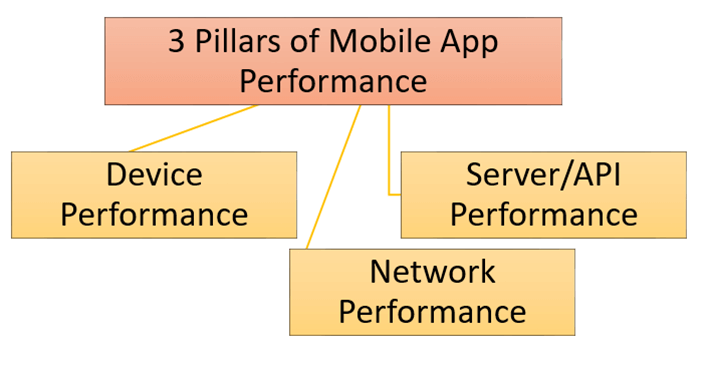
In this tutorial, you will learn-

* [Mobile Application Testing Strategy](http://www.guru99.com/mobile-app-performance-testing-strategy-tools.html#1)
* [Device Performance](http://www.guru99.com/mobile-app-performance-testing-strategy-tools.html#2)
* [Server Performance](http://www.guru99.com/mobile-app-performance-testing-strategy-tools.html#3)
* [Network Performance](http://www.guru99.com/mobile-app-performance-testing-strategy-tools.html#4)
* [Troubleshooting Mobile Applications Performance](http://www.guru99.com/mobile-app-performance-testing-strategy-tools.html#5)
* [Useful Mobile App Testing Tools](http://www.guru99.com/mobile-app-performance-testing-strategy-tools.html#6)
* [Challenges](http://www.guru99.com/mobile-app-performance-testing-strategy-tools.html#7)
* [Set up Mobile App Performance Test Environment](http://www.guru99.com/mobile-app-performance-testing-strategy-tools.html#8)
* [Performance Checklist for Mobile Apps](http://www.guru99.com/mobile-app-performance-testing-strategy-tools.html#9)

**Mobile Application Testing Strategy**

Application performance on a mobile phone or any smart device is usually measured in following three categories.

* Device Performance
* Server/API Performance
* Network Performance

[](http://cdn.guru99.com/images/2-2017/042217_1113_MobileAppPe1.png)

**Device Performance**

When the client experiences slow app, they get annoyed.

For device performance, you will check following -

* **App Start Up**

How much time your app takes to start up? It is the first performance parameter adjudged by the user. As a thumb rule, after the user taps on app icon the first screen should be shown in 1-2 seconds.

* **Battery Time while using an app**

On constant use, some mobile apps, consume a high amount of battery life and heat the phone. This factor adds a lot to the performance of any mobile app and could normally happen when your app is using more resources than required. Excessive resource usage creates a burden on the processor and phone gets heat up.

* **Memory Consumption**

When testing an app, the memory consumption by an app should be checked. By implementing certain functionalities in the app, the memory consumption also increases. For example, in Android apps when push notifications are implemented then memory consumption increases.

In some cases, it has been observed that memory usage by whole O.S is mere 14%, but a new app is consuming 11%. So, these factors must be handled before deploying the app to the real world or giving to the client.

* **Hardware/Software Variation**

When testing a mobile app, it is mandatory to check apps on different devices. It could be the case that app is running smooth on one device but not on other. Like for different vendors of Android devices, we can check the app on Samsung, HTC, and Lenovo phones. Similarly, the app needs to be tested with different RAM and processor specifications like 1 GB or 2 GB.

* **Usage with Other Apps**

When the app under test is running in parallel with other apps, there should be no interference. The best way to check it is by switching app under testing and other apps.

* **App in background**

An app that is running in the background is retrieved, it should remain in the same state as it was before. If this scenario is not handled properly, then data get lost. Again you have to enter data from scratch upon retrieving the app.

**Server/API Performance**

When the app is interacting with the server via API, the response time becomes critical to performance. For Server performance, you will check -

* **Data to and from server**

The app should handle data efficiently that is sent from the server. It must not take too much time while loading data. In certain apps, data is sent in a specified format. So before displaying it in the app, it should be converted to a relevant format. In this process, apps sometimes become slower and response time becomes longer.

* **API Calls Generated from App**

The number of calls from App under test to the server generated from app should be less. In some cases, multiple API calls are made for the same functionality. For better performance, this should be handled with less number of calls.

* **Server Down Time**

Due to any reason if the server is down or unreachable we can save data in the native database. So, whenever the server is down, we can show data stored in the native database. Another solution could be the failover database servers i.e. if one of the servers is down or in maintenance phase the backup server should be available to switch over. The failover/backup server should be in continuous replication and synchronization with the main server.

**Network Performance**

The performance of the app on different networks and network properties need to be measured.

For Network performance, you will check following things.

* **Jitters**

When there is a delay in receiving information on the network, then it is termed as jitters. It is a problem with the connectionless networks or packet switch networks. As the information is distributed into packets, packets can travel by a dissimilar path from the sender to the receiver. When data arrives at the intended location, it becomes scrambled than it was originally sent. In the case of Jitters, the mobile app should be capable enough to handle it.

You need to Show the appropriate notifications to the end user, either to resend the request or wait till the system responds again.

* **Packet Loss**

In the case of complete packet loss, the app should be able to resend the request for the information or should generate the alerts accordingly. If data is not complete, then the user will not be able to comprehend information displayed in App. This can be stressful for the user. So, it is better to display a suitable message or prompt user to try again.

* **Network Speed**

The app needs to be checked on a variety of networks with variable speed. The app should be tested on 2.5G, 3G, and 4G networks. Both Wi-Fi and mobile networks are included in this. Also, the behavior of app should be monitored. Especially, when both networks are available, and switching occurred from one network to another.

For example, an issue may arise in an app for the users while switching phone network from 4G to WIFI and vice versa. In this case, the app becomes unresponsive and may require restarting the app for use.

**Troubleshooting Mobile Applications Performance**

After discovering the issues/problems while performance testing. It is time to trace and correct faults.

**Problem 1) Lag or sluggish response of the Mobile App.**

The cause of this delay may be the RAM, Cache, etc.

You need to kill unnecessary processes or clear the cache. Troubleshooting the connectivity issue may solve some of the problems that are creating lags

**Problem 2) App Restarting, locking up, freezing or unresponsiveness.**

It may be fixed by some of the following steps

* Optimizing the application codes
* Software should be patched and updated.
* Automatic restores
* Managing RAM or in some cases ROM while using external cards
* Wiping the cache partitioning
* Verifying the app working with other third party apps and API's
* Mapping the mobile application according to device

**Useful Mobile App Testing Tools**

Mobile app testing tools vary according to the devices or mobile OS. Some common mobile app performance testing tools are

**ANDROID**

* [Robotium](http://robotium.com/)

It is just like Selenium for Mobile Apps. The tester can record and play several steps that are required to perform testing.

* [Monkey Runner](https://developer.android.com/studio/test/monkeyrunner/index.html)

MonkeyRunner can run tests on real devices connected to a PC or emulators. The tool has an API, which allows controlling a smartphone, a tablet or an emulator from outside of Android code.

**APPLE**

* [Automator (Mac)](http://macosxautomation.com/automator/)

Automator is an application developed by Apple for OS X. It implements point-and-click (or drag and drop) creation of workflows for automating repetitive tasks into batches for quicker alteration. This saves time and effort over human intervention to manually change each file separately.

**Challenges**

Key challenges faced while performance testing include

* Organizing different mobile platforms and their operating systems
* Simulating Connectivities like Edge, 3G, 4G or WiFi, etc.
* Mobile devices constraints like battery and resources consumption
* Mobile phone usability
* The assorted sizes of mobile devices to run the same app

**Set up Mobile App Performance Test Environment**

To configure Test Environment, you need to-

* Understanding of the mobile app which needs to be tested
* Identification of different OS on which the app needs to run
* Building the test setup
  + Build the emulators or simulators
  + Prototyping of the actual setup
* Selecting the appropriate tool for the testing

**Mobile App Performance Testing Checklist**

Testing the performance of the mobile apps is an important measure before release. Performance testing is done to check

* How much of the RAM is required for utilizing this app?
* To verify speed and response time of APP under different networks and circumstances.
* Ensure realistic user experience under several network conditions
* Ensure the required results are achieved in case of multiple connectivities
* Ensure the application do not get crashed.
* Ensuring the mobile applications perform well while using data, Wi-Fi or other connectivity
* Monitoring the uptime and the mobile API usage bottlenecks
* To ensure the maximum number of simultaneous users
* Finally, to check the mobile app to its limits

**Summary**

* Performance testing requires an understanding of Mobile App, resource utilizer, virtual users, emulators and multiple test strategies.
* App performance on a mobile phone is measured in following three categories.
  + Device Performance
  + Server Performance
  + Network Performance
* Performance testing challenges include compact sizes of the mobile devices, resources availability, costing, and budgeting.