



Mobile Testing

tutorialspoint
SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

This tutorial will help the audience to learn the different aspect of the up-trending mobile device testing as well as mobile application testing. You will get familiar with many useful tools for black-box and white-box testing of a mobile application.

This tutorial also provides a deep insight on mobile device automation testing. Using this tutorial, you can enable yourself for up-to-date test planning for mobile device and mobile device application testing. In addition, you shall be able to automate basic test scripts for mobile device application testing.

Audience

If you are a quality assurance engineer having interest in mobile device testing as well as mobile device application testing, this tutorial will turn out to be a helping guide.

Prerequisites

A reader should know basic software testing concepts such as test planning, black-box testing tricks, etc. In addition, it will help a great deal if the reader is familiar with any scripting languages, for example, JavaScript.

Disclaimer & Copyright

© Copyright 2019 by Tutorials Point (I) Pvt. Ltd.

All the contents and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book can download, read, print, or keep it for his/her personal use. However, it is strictly prohibited to reuse, retain, print, copy, distribute, or republish whole or the part of this e-book in any manner for the commercial purpose without written consent of the publisher.

We strive to produce and update the contents and tutorials of our website accurately and precisely, however, the contents may contain some inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. takes no guarantee regarding the accuracy, timeliness, or completeness of our website or its contents including this tutorial. If you find any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com.

Table of Contents

About the Tutorial.....	1
Audience	1
Prerequisites	1
Disclaimer & Copyright.....	1
Table of Contents	2
1. MOBILE TESTING – OVERVIEW.....	6
2. MOBILE PLATFORMS	7
Mobile Operating Systems	7
3. MOBILE DEVICE TYPES.....	8
4. NATIVE VS HYBRID VS MOBILEWEB.....	9
Mobile Web	9
Native App	10
Hybrid App	10
5. DEVICE VS APPLICATION TESTING	12
Device Testing	12
Application Testing	12
6. MOBILE EMULATOR VS SIMULATOR.....	14
7. MOBILE APPLICATION TESTING	16
Functional Testing	16
Compatibility Testing	16
Localization Testing.....	16
Laboratory Testing	17
Performance Testing	17

Stress Testing	17
Security Testing	18
Memory Leakage Testing	18
Power Consumption Testing	18
Interrupt Testing	19
Usability Testing.....	19
Installation Testing.....	20
Uninstallation Testing	20
Updates Testing	20
Certification Testing	21
8. MOBILE UI TESTING	22
Screen Orientation / Resolution	22
Available Tools	23
Touch Screens	26
Soft & Hard Keys	26
9. MOBILE DEVICE TESTING – PLANNING AND TOOLS.....	28
Preparing to Perform Mobile Device Testing	28
Device Management Tools	28
10. MOBILE DEVICE TESTING – HARDWARE PERSPECTIVE	31
Hardware Components	31
UDID for iOS	34
Provision Profile for iOS	34
Device Options & Preferences.....	35
11. MOBILE DEVICE TESTING – TYPES	36
Network Connections.....	36

SD Card Interactions.....	36
Bluetooth Testing.....	38
Wi-Fi Testing	38
Localization & Globalization	40
Database Testing.....	41
Recovery Testing	41
Concurrency Testing.....	42
Usability Testing.....	42
12. TESTING FRAMEWORKS – OVERVIEW	44
13. ANDROID TESTING FRAMEWORKS.....	45
14. IOS TESTING FRAMEWORKS	46
15. ROBOTIUM TESTING FRAMEWORK.....	47
Benefits of Robotium	47
Drawbacks of Robotium	47
How to Use Robotium	48
Test an App with Robotium.....	51
16. SELENDROID TESTING FRAMEWORK	54
Benefits of Selendroid	54
Drawbacks of Selendroid.....	54
How to Use Selendroid	54
Test an App with Robotium.....	56
17. APPIUM TESTING FRAMEWORK	60
Advantages of Appium	60
Drawbacks of Appium	60

Test an App with Appium	64
18. ZUCCHINI TESTING FRAMEWORK.....	67

1. Mobile Testing – Overview

Before we start the actual tutorial, let's have some fun. Take a look at the following list. You can easily relate to these essential chores that we invariably have to face in today's busy and hectic lifestyle:

- I want to pay my electricity bill.
- I need to communicate with my manager and submit my reports urgently.
- I want to buy new clothes for my kid, but I don't have the time to go to a store.
- Oh...it's 10 PM; I am running late for my flight. Where is my cab?
- I have been in this city for the first time; which hotel should I book?

Now answer yourself. What do you need to perform these activities in a fraction of second? The answer would be:

- A smartphone,
- Internet connectivity, and
- A mobile app to do the job.

This makes us realize the importance of a mobile app in today's era. Everything is being done in a smart way through your smart phone. Each day, we get to know about a new app or tool being launched to ease our life.

There is one well-known incidence about **Gowalla**. Millions of people enjoyed this location-based social network that started in 2007 and shuttered down five years later. Multiple problems blocked Gowalla from reaching mass appeal. One of the prime reasons that surfaced was "Checking in was not that user friendly".

To have a successful mobile app, we need to understand that developing a nice mobile app is not the only requirement. A mobile app needs to be tested thoroughly in order to get the likability of users. After all, it is the user experience makes any software successful.

2. Mobile Platforms

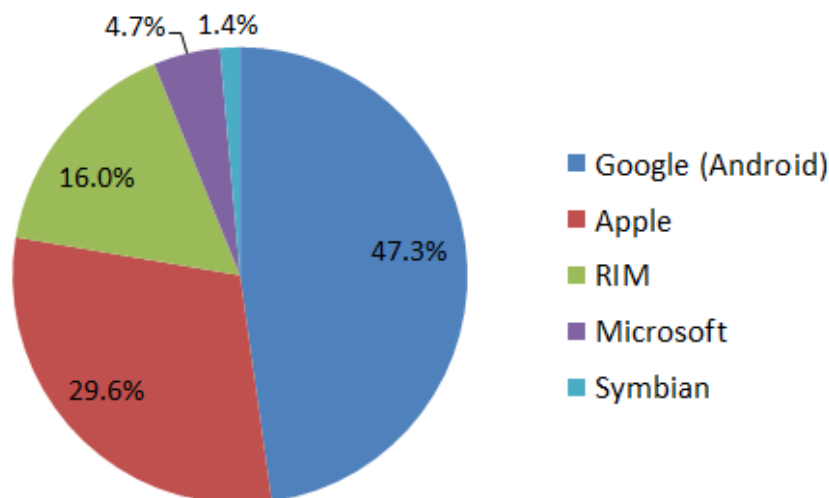
Before jumping on to mobile testing, it's advisable to understand the basics of mobile platforms. It mainly includes the operating system of the mobile, type of the device, and type of the mobile app. Having sufficient knowledge about each of these will help us conduct robust test planning in the long run.

Mobile Operating Systems

The following table gives an overview of some of the popular mobile operating systems available in market:

Operating System	Developed by	Popularity (Low, Medium, High)	Latest available version
Android	Google Inc	High	Lollipop, Android 5.0-5.1
iOS	Apple Inc	High	iOS 8.X
Blackberry	Blackberry Ltd	Low	Blackberry 10.2.1
Windows	Microsoft Inc	Medium	Windows 10 Mobile
Symbian	Symbian Foundation	Low	Discontinued

Based on some generic survey, the usage of different operating systems in the market can be depicted as shown below.



3. Mobile Device Types

Mobile devices are typically handheld computers. They have many variants based on their characteristics such as physical dimension, hardware and software capability, what are they meant for, etc.

Take a look at the following table. It differentiates tablets, e-book readers, and smartphones based on their characteristics.

Device	Tablets	E-book Readers	Smartphones
What it is	Tablets are portable computer devices. Unlike traditional computers, they don't have keyboards or mouse, however the entire screen is touch sensitive.	E-book readers—also called e-readers—are similar to tablet computers, except they are mainly designed for reading e-books (digital, downloadable books).	A smartphone is a powerful mobile phone that is designed to run a variety of applications in addition to providing phone service.
Used for	Almost all the jobs which we can do with traditional computers or desktops.	Reading e-books	Web browsing, watching videos, reading e-books, and playing games
Example	Samsung Tablets	Amazon Kindle, Barnes & Noble Nook.	Sony smartphones, Samsung smartphones, Apple iPhone.

4. Native vs Hybrid vs MobileWeb

A critical factor that you have to consider while doing test planning is checking the Mobile Application type. You will mainly come across three types of mobile applications: Mobile Web, Native App, and Hybrid App. The classification is based on the development efforts and App redistribution strategy. Let's understand each of them in detail.

Mobile Web

Web apps are not real applications; they are actually websites that open in your smartphone with the help of a web browser. Mobile websites have the broadest audience of all the primary types of applications.

- **Example:** <http://www.tutorialspoint.com/>
- **Benefits :**
 - Easy access.
 - Easy Development: Developing responsive design and restructuring the content to be properly displayed on a smaller screen/hardware will make any desktop website mobile friendly.
 - Easy update: Just update in one location and all the users automatically have access to the latest version of the site.
 - No installation required, as compared to native or hybrid app.
- **Downside :**
 - Mobile websites cannot use some of the features. For example, access to the file system and local resources isn't available in websites.
 - Many existing websites don't support offline capabilities.
 - Users won't have the app's icon on their home screen as a constant reminder. The website needs to be opened in a web browser only.
 - While native and hybrid apps appear on the App Store and Google Play, web apps won't. So redistribution is not that sensible.

Native App

A native app is developed specifically for one platform. It can be installed through an application store (such as Google Play Store or Apple's App Store)

- **Example:** Whatsapp, Facebook
- **Benefits :**
 - Native Apps live on the device and are accessed through icons on the device home screen.
 - They can take full advantage of all the device features — they can use the camera, the GPS, the accelerometer, the compass, the list of contacts, and so on. They can also incorporate gestures (either standard operating-system gestures or new, and app-defined gestures).
 - Native apps can use the device's notification system and can work offline.
 - Publishers can make use of push-notifications, alerting users every time a new piece of content is published or when their attention is required.
 - Native Apps maintain UI design of each operating system, thus they offer the best user experience. For example, a Native App can have a left-aligned header in Android and a center-aligned header in iOS.
 - Redistribution is easy, as it is found in app store.
- **Downside :**
 - High cost for building the app : Native apps developed for one platform will not run on another platform. An App built for Android will not run on iOS. We need to build a different App altogether for iOS. Because of this reason, we need to maintain multiple versions of the App.
 - Even though you might publish native Apps, you'll want to keep the mobile website well maintained, as mobile brings more traffic. So maintenance is higher.

Hybrid App

Hybrid Apps are a way to expose content from existing websites in App format. They can be well described as a mixture of Web App and Native App.

- **Example :** Instagram, Wikipedia
- **Benefits :**
 - Developing a Hybrid App is cheaper than developing a Native App. It can be built for cross-platforms, i.e., reduced cost for App development.

- Maintenance is simple, as there are not many versions to be maintained.
 - It can take advantage of a few features available in the device.
 - It can be found in the App Store, which makes the distribution easy.
 - It has a browser embedded within the app only.
- **Downside :**
 - Graphics are less accustomed with the operating system as compared to Native Apps.
 - Hybrid Apps are slower than Native Apps.

5. Device vs Application Testing

Device Testing

This type of testing is generally being done to ensure the quality of a mobile device. Testing includes hardware and software testing for a mobile. We will discuss here the different types of testing which are generally being carried out on a mobile devices.

Unit Testing

Unit testing is a test phase when portions of a mobile device development are tested, usually by the developer. It may contain hardware testing, software testing, and mechanical testing.

Factory Testing

Factory testing is a kind of sanity check on mobile devices. It is conducted automatically to verify that there are no defects brought by the manufacturing or assembling. It mainly includes the following testing:

- Mobile application testing
- Hardware testing
- Battery (charging) testing
- Signal receiving
- Network testing
- Protocol testing
- Mobile games testing
- Mobile software compatibility testing

Certification Testing

Certification testing is the check before a mobile device goes to the market.

Application Testing

Mobile application testing is a process by which application a software developed for handheld mobile devices is tested for its functionality, usability, and consistency. There are different types of testing which can be performed on a mobile device. For example,

- Functional Testing
- Laboratory Testing
- Performance Testing
- Memory Leakage Testing
- Interrupt Testing

- Usability testing
- Installation testing
- Certification Testing
- Security Testing

Key Points

- Device testing is generally being carried out to check the mobile device itself, whereas Mobile application testing involves testing of an application which will be running on the chosen device.
- When we call it device testing, hardware testing becomes a part of it. In case of mobile application testing, it depends, i.e., if the application under test requires hardware integration, then hardware testing will be involved.
- Automation is possible for Mobile Device testing as well as for Mobile application testing.

6. Mobile Emulator vs Simulator

One thing is self-explanatory in case of mobile testing. To perform mobile testing, you need a mobile device. This is to access that how our product will work and look like on a given mobile set.

Suppose we are developing an application for flight ticket booking system. Once the product is entirely developed, as a part of mobile testing, we need to check if the application is working as expected with all the majorly used devices like Android phones, iOS, Blackberry phones, and other different types of tablets and iPads.

To do this kind of check, we need to acquire each such device and then we can check if the application behaves as per expectation. Yes you thought right, as a product owner one will defiantly find this very expensive to procure such a large number of mobile devices and carry out testing. So is there any smart alternate available?

The solution to this problem is to use Mobile Simulators and Mobile Emulators. These are primarily software programs designed to provide simulation for important features of a smartphone. They are very similar in nature, so sometimes, they are used interchangeably.

Let's compare how testing on an Emulator/Simulator is different from testing on a real device:

	Real Device	Emulator / Simulator
Price	Getting real devices will cost you a lot.	It is almost free, we just need to download and install them
Processing Speed	It has faster processing; however network latency may be normal.	It is slower as compared to actual devices. It has observed less latency than real devices connected to the local network or in the cloud.
Debugging	Debugging is not that easy.	It provides step-by-step debugging of an application. Also, it provides an efficient way for capturing screenshots.
Web-app Testing	Web applications can be tested in a normal way.	Testing a web application is much easier.
Reliability	Testing on a real device has a major advantage that it always gives accurate results.	It cannot simulate all types of user interactions; hence it may lead to false results sometimes. So it scores low when it comes to reliability.

A simulator/emulator cannot mimic the following features:

- Mobile device battery
- Mobile device's camera
- Difficult to mimic interruptions like incoming calls and SMS.
- Not so much realistic simulation for mobile device memory usage.

Let us now understand more about mobile simulators and mobile emulators. There are specific differences between the two. The following table lists the major difference between a simulator and an emulator.

	Emulator	Simulator
What it mimics	<ul style="list-style-type: none"> - Mobile device software - Mobile device hardware - Mobile operating system 	<ul style="list-style-type: none"> - Internal behavior of the device. - It does not mimic hardware.
How to get it	It is generally provided by the device manufacturer.	It is generally provided by the device manufacturer or some other company.
Internal structure	It is written in machine-level assembly language.	It is written in high-level language.
Debugging	It is more suitable for debugging.	It is not suitable for debugging purpose.
Performance	Emulators are really slow. Emulating the actual hardware usually makes the software run slower than it would natively.	Faster than emulators.
Example	Google's Android SDK	Apple's iOS Simulator

So, what is the best choice for mobile testing? Best practice indicates that, while actual development is in progress, we should use an emulator or a simulator. Before finalizing the product, there should be a sanity check with chosen real devices. For example, there is a huge number of Android smartphone users, so the smart choice is to have a sanity check for the latest Android device and regression can be conducted over simulators.

7. Mobile Application Testing

A simple definition of mobile application testing would go like this “**Mobile application testing** is a process by which an **application software** developed for handheld mobile devices is tested for its functionality, usability, and consistency. Mobile application testing can be automated or manual type of testing.”

Note: For better understanding, we shall assume that we are testing a mobile application for online flight ticket booking system.

Functional Testing

Functional testing is the most basic test for any application to ensure that it is working as per the defined requirements. Similar to other user-interface based applications, mobile applications require a number of human interactions in user scenarios.

Example test scenarios:

- Verify that flight availability is displayed for a chosen source destination on a selected date only.
- Verify that past dates are not included in the search results.

Compatibility Testing

Compatibility testing has got the highest stack when it comes to mobile application testing. The purpose of a mobile app compatibility test, in general, is to ensure an app's key functions behave as expected on a specific device. The compatibility itself should only take a few minutes, and can be planned well in advance.

It is not going to be an easy task, deciding on which mobile devices compatibility tests should be performed (as testing with all available devices is merely impossible). So prepare a test matrix with every possible combination and get it prioritized by the client.

Example test scenarios:

- Verify that flight search is performed successfully with Android device.
- Verify that flight search is performed successfully for Apple iPad.

Localization Testing

Nowadays, most of the apps are designed for global use and it is very important to care about regional trails like languages, time zones, etc. It's important to validate the app's functionality when someone changes the time zone. It has to be taken into consideration that sometimes western designs may not work with the audience from eastern countries or vice-versa.

Example test scenarios:

- Verify that there is no UI or data truncation issue when we use the mobile app with different languages (or say, non-English language).
- Verify that time zone changes are handled gracefully for your mobile application.

Laboratory Testing

Laboratory testing, usually carried out by network carriers, is done by simulating the complete wireless network. This test is performed to find out any glitches when a mobile application uses voice and/or data connection to perform some functions.

Example test scenarios:

- Verify that there is no glitch while a customer is having voice chat with the support staff.

Performance Testing

Mobile performance test covers client application performance, server performance, and network performance. It is important to make sure that the performance test scenarios cover all those areas. With the help of performance testing tools, it is not difficult to identify the existing networks, servers, and server-side application bottlenecks, given the predefined loads and transaction mix.

Example test scenarios:

- Verify that flight available check takes only a reasonable amount of time.
- Verify that while flight availability is being checked, the mobile operates normally and does not hang.

Stress Testing

Stress testing is a must to find exceptions, hangs, and deadlocks that may go unnoticed during functional and user interface testing. Here is a list of some of the criteria for stress testing:

- Load your application with as much data as possible to try to reach its breaking point.
- Perform the same operations over and over again.
- Perform the repeated operations at varying speeds – very quickly or very slowly.
- Leave your application running for a long period of time, both interacting with the device and just letting it sit idle, or performing some automatic task that takes a long time, for example, a slideshow.
- Randomly send screen taps and keystrokes to your application.

- Have multiple applications running on your device so that you can switch between your application and other device applications often.

Example test scenarios:

- Check 1000 users are accessing the mobile app to search for domestic flights.
- Check 1000 users are accessing the mobile app to search for international flights.

Security Testing

Vulnerabilities to hacking, authentication, and authorization policies, data security, session management and other security standards should be verified as a part of mobile app security testing. Applications should encrypt user name and passwords when authenticating the user over a network.

One way to test security related scenarios is to route your mobile's data through a proxy server like OWASP Zed Attack Proxy and look for vulnerabilities.

Example test scenarios:

- Verify that the application does not get operated with same user credentials on two different mobile devices.
- Verify that a session automatically gets expired if it remains inactive for more than 15 minutes.

Memory Leakage Testing

Mobile devices have very limited memory as compared to other computers, and mobile operating systems have a default behavior to terminate applications that are using excessive memory and causing a poor user experience.

Memory testing is exceptionally important for mobile applications to ensure that each application maintains optimized memory usage throughout the user journey. It is recommended that we conduct memory testing on the actual target device, since the system architecture is different from an emulator to an actual device.

Example test scenarios:

- Do flight availability checks ten times and note down increasing memory usage for each check.
- Keep the application running for ten minutes and observe if memory usage remains stable.

Power Consumption Testing

There are several types of batteries used in different mobile devices (i.e. nickel cadmium/ lithium ion/ Nickel metal hybrid). While we focus on power consumption testing, we are

required to measure the state of the battery at each activity level. It will give us a better understanding of power consumption by an individual application.

Power Consumption test can be done manually; also there are some free tools available in the market such as Trepn Profiler, Power Tutor, and Nokia Energy Profiler. These are applications which can display the real-time power consumption on a smartphone or tablet.

Example test scenarios:

- Use the mobile app to search for flight availability, and check the power consumption remains minimal.
- Keep the mobile app in ideal condition; verify that there is no power consumption when there is no activity happening for the app.

Interrupt Testing

An application, while functioning, may face several interruptions like incoming calls or network coverage outage and recovery. This can again be distinguished for:

- Incoming and Outgoing SMS and MMS
- Incoming and Outgoing calls
- Incoming Notifications
- Battery Removal
- Cable Insertion and Removal for data transfer

Example test scenarios:

- Verify that flight availability check gets paused and resumed back after receiving an incoming call.
- Verify that the user can reject a call while using the App and again resume the same App thereafter.

Usability Testing

Usability testing evaluates the application based on the following three criteria for the target audience:

- **Efficiency:** The accuracy and completeness with which specified users can achieve specified goals in a particular environment.
- **Effectiveness:** The resources expended in relation to the accuracy and completeness of goals achieved.
- **Satisfaction:** The comfort and acceptability of the work system to its users and other people affected by its use.

It is very important to get usability testing in place from the early stage of the application design, and it should not be done only when the application is completed. Usability testing requires heavy involvement of users, and the output might affect the application design, which is very difficult to change in the later stages of the project.

Example test scenarios:

- Flight availability check should be on the home page.
- Sponsored advertisement should not be displayed in the middle of the content.

Installation Testing

Installation testing verifies that the installation process goes smoothly without the user having to face any difficulty.

Example test scenarios:

- Verify that the installation process is smooth and does not take long.
- Verify that the installation is successful through enterprise app store.

Uninstallation Testing

The basics of uninstallation testing can be summarized in one line as “Uninstallation should sweep out data related to the App in just one go”.

Example test scenarios:

- Verify that all the application related files are removed successfully after uninstallation.
- If it is an App which stores media files (like Whatsapp or Facebook), retain the files even after the uninstallation of the App.

Updates Testing

We need to be very much cautious about mobile app updates. People frequently complain about applications not working satisfactorily after an update. So it is very important that under the update testing, we qualify that the App will work as it was working previously. In a nutshell, it should not break anything. Mobile application updates can take place in two ways – **Automatic update** and **Manual update**.

Example test scenarios:

- Verify that the application works successfully after automatic update.
- Verify that the update progress is displayed properly.

End of ebook preview

If you liked what you saw...

Buy it from our store @ <https://store.tutorialspoint.com>