Table of Contents

1. Mobile Application Testing Checklist

here will be all mobile testchecklist
for any app
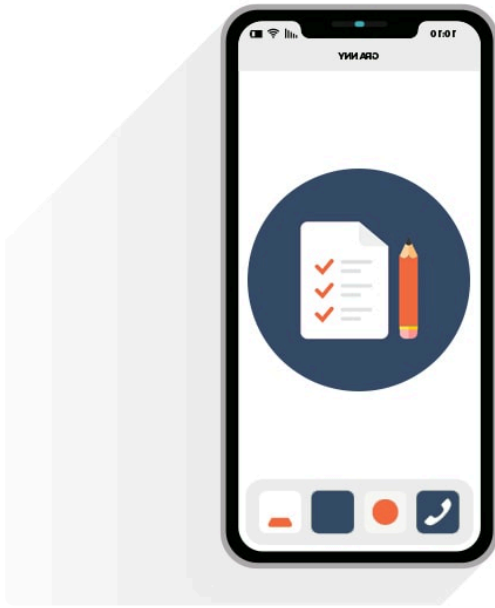
2. 1. DEVICE SPECIFIC CHECKS

1.1 Can the app be installed on the device?

1.2 Does the app behave as designed/desired if there is an incoming call?

1.3 Does the app behave as designed/desired if there is an incoming SMS?

1.4 Does the app behave as designed/desired if the charger is connected?

1.5 Does the app behave as designed/desired if the charger is disconnected?

1.6 Does the app behave as designed/desired if the device goes to sleeping mode

1.7 Does the app behave as designed/desired if the device resumes from sleeping mode

1.8 Does the app behave as designed/desired if the device resumes from lock screen?

1.9 Does the app behave as designed/desired if the device is tilted?

1.10 Does the app behave as designed/desired if the device is shaken?

1.11 Does the app behave as designed/desired if a local message is coming from another app (think of: calendar reminders, to-do task etc.).

1.12 Does the app behave as designed/desired if a push message is coming from another app (think of: twitter mentions, whatsapp message, wordfeud invitation, etc).

1.13 Does the app interact with the GPS sensor correctly (switch on/off, retrieve GPS data)?

1.14 Is the functionality of all the buttons or keys on the device defined for this app?

1.15 Verify that buttons or keys which have no defined function have no unexpected behaviour on the app when activating.

1.16 In case there's a true "back" button available on the device does the "back" button take the user to the previous screen?

1.17 In case there's a true "menu" button available on the device, does the menu button show the app's menu?

1.18 In case there's a true "home" button available on the device, does the home button get the user back to the home screen of the device?

1.19 In case there's a true "search" button available on the device, does this get the user to some form of search within the app?

1.20 Does the app behave as designed/desired if the "Battery low" message is pushed

1.21 Does the app behave as designed/desired if the sound on the device is turned off?

1.22 Does the app behave as designed/desired if the device is in airplane mode?

1.23 Can the app be de-installed from the device?

1.24 Does the application function as expected after re-installation?

1.25 Can the app be found in the app store? (Check after go-live)

1.26 Can the app switch to different apps on the device through multitasking as designed/desired?

1.27 Are all touch screen positions (buttons) working when a screen protector is used.

3. 2. NETWORK SPECIFIC CHECKS

2.1 Does the app behave according to specification if connected to the internet through Wi-Fi?

2.2 Does the app behave according to specification if connected to the internet through 3G?

2.3 Does the app behave according to specification if connected to the internet through 2G?

2.4 Does the app behave according to specification of the app is out of network reach?

2.5 Does the app resume working when it gets back into network reach from outside reach of the network?

2.6 Update transactions are processed correctly after re-establishing connection.

2.7 Does the app still work correctly when tethering or otherwise connected to another device

2.8 What happens if the app switches between networks (Wi-Fi, 3G, 2G)

4. 3. APP SPECIFIC CHECKS

3.1 Has the app been tested on different type of devices and different versions of OS?

3.2 Stability check: if the app has a list (for instance of pictures) in it, try scrolling through it at high speed.

3.3 Stability check: if the app has a list (for instance of pictures) in it, try scrolling to before the first picture or behind the last picture.

3.4 Is downloading of the app prevented in case it's bigger than the OS allows downloading when connected to cellular networks.

3.5 Integration: does the app connect correctly to the different social networks (LinkedIn, twitter, facebook, etc).

3.6 The app does not interfere with other apps when in background/multitasking mode (using GPS, playing music, etc.).

3.7 Can the user print from the app (if applicable)

3.8 The search option in the app displays relevant results

3.9 Verify most common gestures used to control the app.

3.10 What happens if you select different options at the same time (undesired multitouch, for example – select two contacts from the phone book at the same time).

3.11 App name should be self explanatory

3.12 Does the app limit or clean the amount of cached data.

3.13 Reloading of data from remote service has been properly designed to prevent performance issues at server-side. (manual reloading of data can reduce the amount of server calls)

3.14 Does the app go to sleep mode when running in the background (prevent battery drain)

5. 4. APP USER INTERFACE CHECKS

4.1 To keep controls as unobtrusive as possible for instance by fading them out if they are not used for a while.

4.2 Make it possible for users to go back to a previous screen for instance by adding a back or cancel button

4.3 The main function of the app should be apparent immediately. It should speak for itself.

4.4 Use at most one action on the screen that is highlighted as the most likely for the user. (Example: in iOS a blue button represents the default or most likely action).

4.5 Minimize user actions by using a picker or a table view where users can select a certain choice over a data entry field where users have to type a choice

4.6 In an app, the user should not be able to store files locally, outside the app sandbox.

4.7 In an app, the user should not be exposed to the permissions of a specific file

4.8 If there is a long list of data to scroll trough, provide a search option above the list.

4.9 If performance is slow, indicate a progress status icon ("Loading…"), preferably with specific message.

4.10 In case of 'live' filtering of data while the user enters his search query, verify the performance.

4.11 The appearance of buttons that perform standard actions are not altered in the app (for instance: refresh, organize, trash, Reply, back, etc.)

4.12 Do not use standard buttons for other functions then that they are normally used for

4.13 The app should respond to all changes in device orientation, as per the design

4.14 Do not redefine gestures in your app that have a standard meaning (example: swiping from top to bottom enables the notification center)

4.15 Requirement to login is delayed in the app as long as possible

4.16 If the app is stopped at an unexpected time, user data should be saved locally and available at start-up.

4.17 Users should be warned of the consequences of deleting a document

4.18 Keyboard adjusts to expected input (for instance numbers/letters when expected).

4.19 Are inactive buttons clearly distinguished from active buttons?

6. 5.Functional test to check if the App meets its purpose

1. App's performance for designated and non-designated tasks
2. Compatibility in different devices, screen size, resolutions, OS and browsers.
3. Integration with other services like location/GPS, Wi-Fi, and social media
4. Functioning of redirect options
5. Application consistency in its desktop version
6. Adequate prevention from performing undesired actions
7. Misinterpretation or comprehensibility issues on the UI and design of the app.

7. 6.Security and Data Privacy for Authenticated App usage

1. Customers' Payment data security
2. Network protocols security for running applications
3. Breach in applications' security and error reporting
4. Authenticating application certificates and permissions
5. Automatic Application lock out upon continuously entering invalid

8. 7.Performance testing

1. Time taken to launch the application
2. Application performance at times of peak load conditions and continual key pad entry
3. Splash performance check and ensuring it remains on screen for less than 3-4 seconds
4. Application performance in low device battery and charging conditions
5. Deploy Live Monitoring Services to keep the app's computing power on check
6. Integration with the device and other applications without hampering performance.
7. Application install/uninstall successfully within the desired timeframe
8. Displaying error messages and exits gracefully at times of low memory issues
9. App Performance at the time of a Network problem and error alert prompts
10. Application performance when the Network is back in action

9. 8.App performs well in the background

1. Making/Receiving voice calls while the app is running
2. Rejecting calls while viewing the application
3. Resuming App from the same point as left at the time of switching to another app
4. Sending/Receiving messages and alerts while the app is running
5. Sending/Receiving Push Notifications and responding to them