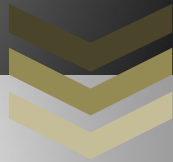


Mobile Testing: Ready Reckoner

**Ajay Balamurugadas &
Sundaresan Krishnaswami**



Mobile Testing: Ready Reckoner



Mobile Testing: Ready Reckoner

Foreword

I love mobile devices and apps. Right from my first BlackBerry, followed first by an iPhone, then a Droid for short while and back to iPhones since then, I have always loved them. Some would say way too much. But through all that usage, and all the acclaimed simplicity of some of the OS'es, one thing which has been very clear to me is that there is nothing intrinsically simple about the technology that makes these mobile devices what they are. That this is a challenge which a lot of us relish and live every single day doesn't still make it simple.

So when Sundar and Ajay, talked to me about their idea of a Ready Reckoner for Mobile Testing, I was certainly intrigued. Both of them have left me impressed many times with the thoroughness of their testing that goes beyond the obvious. And as we have learnt from battery challenges, incorrect locations, crashes on accessing calendar and myriad other issues being reported from the field, there is a very clear need to bring a little more method, little more science to the art of testing apps on mobile devices.

Mobile Testing: Ready Reckoner

Sundar and Ajay have clearly delivered a solid reckoner that is part checklist, part knowledge bank and part guide/manual on how to do mobile testing. If nothing else, this will help all testers to develop their skills on the mobile OS, ensure that a lot of known pitfalls are covered and then spend more energy on the real tricky issues. Like that incorrect roaming status being reported on iOS devices :)

Jatin Malik

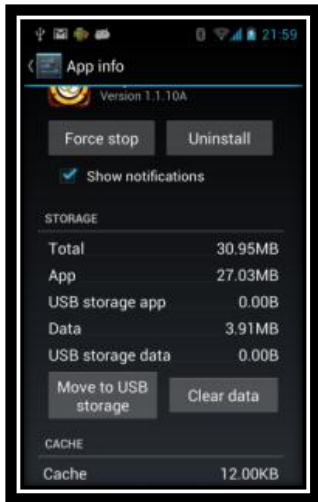
Director, Online Platform Services at MaaS360

Mobile Testing: Ready Reckoner

Terms To Know



Mobile Testing: Ready Reckoner



Install App on SD Card

When an app is installed from the Google Play Store, it is installed on the device memory. However, navigate to Settings > Application > <App> and use the move to SD card option to move the app.

Learning:

- Remove SD Card?
- Any strange behavior when the device is rebooted and the SD card is not mounted?
- What if the app is in use and SD card is un-mounted?
- Does the app exit gracefully when certain directories are deleted from SD card and app is launched?

Mobile Testing: Ready Reckoner

- Try on an encrypted SD card
- Use the same card on another device?

Tools to use:

Use the tool Moborobo for Windows to install apps directly on SD cards

Mobile Testing: Ready Reckoner



App Upgrade

Install an existing app, make required configurations and upgrade it. While upgrading an app, Android replaces existing .jar files if there are any changes and modifies the database files if required. Any unwanted files from the previous version are deleted.

Learning:

- Existing app settings could be wiped out
- User might receive notifications that were already dismissed
- Upgrade could corrupt existing database
- Required dependency files could be deleted
- Encrypted user files may be decrypted during an

Mobile Testing: Ready Reckoner

upgrade and not encrypted again

Tools you can use:

1. You can upgrade an app via command line use the following command
adb install -r <appname.apk)
(-r is not required if the app is not installed already)
2. Add the file to the device's SD card and open it through the file manager. Replace existing file
3. Upgrade it through Google Play
4. Use a tool like Moborobo to install it over existing application.

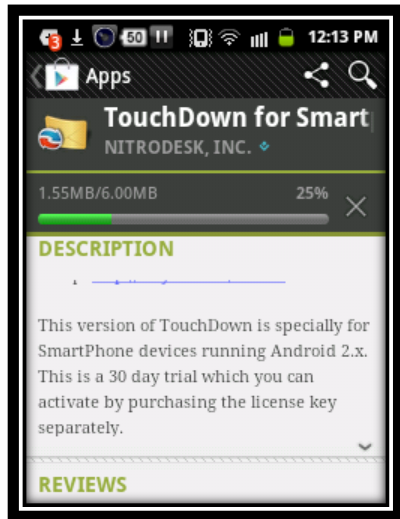
Mobile Testing: Ready Reckoner

Claims Testing

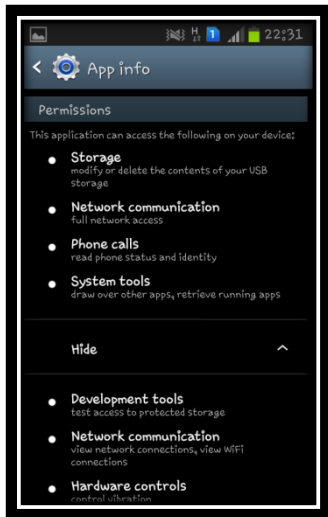
Before you install the app from the Store (Google Play or iTunes App Store), read the description of the app. Check if the claims are fulfilled or not.

Learning:

- Check supported versions and pre-requisite for the app to work properly
- Check the app name, app icon, provider and the app size. Confirm the information after installation
- Is the claim on Google Play on device same as the claims on the app website?



Mobile Testing: Ready Reckoner



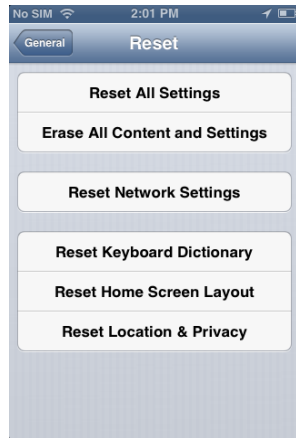
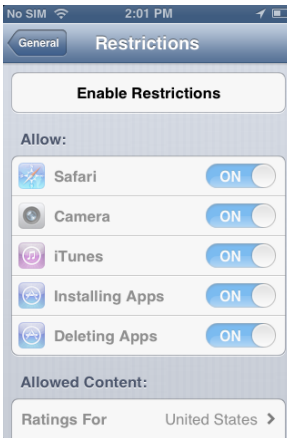
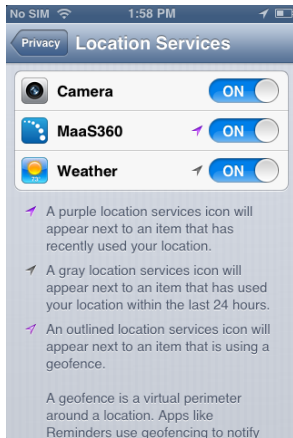
App Permissions

During installation, app should ask the user for permissions to use the device resources or settings. What if you accept and later remove the permissions?

Learning:

- Make a note of the required permissions prompted during installation. After installation, tap on Applications, app info and check if there is any change?
- Try changing the permissions one by one. Example: change encryption on SD card, change the settings of the device from the Settings menu
- On Android and iOS, there are different settings to control these permissions. Some of the settings are highlighted in screenshots here

Mobile Testing: Ready Reckoner



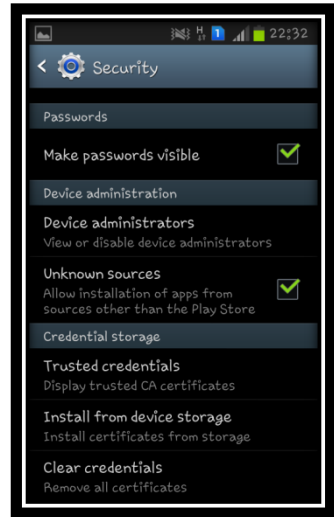
Mobile Testing: Ready Reckoner

Device Administrators

On Android a security application needs to be a device administrator to perform user specified actions. These actions are useful to control the device remotely, should the device be stolen or fall into wrong hands.

Learning:

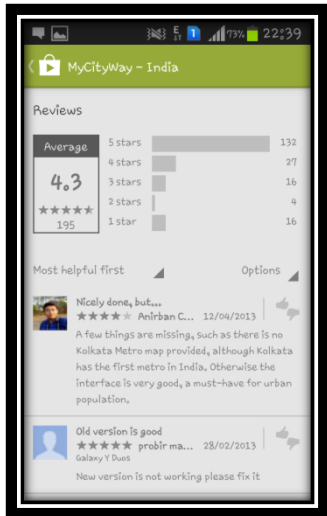
- Device administrator should be an optional feature unless it is an email or MDM application
- App should prompt the user to authorize as a device administrator
- Remote management activity on the device must be communicated securely via SSL, policies applied on the device should be encrypted



Mobile Testing: Ready Reckoner

- If the app supports inter-app communication, ensure that the third party app or service is whitelisted and trusted by the app with device admin controls

Mobile Testing: Ready Reckoner



Review Feedback on Google Play

Simply learn from Google Play feedback

Learning:

- How does your app perform in the wild?
- What are they talking about it?
- Do they post issues instead of reviews? Can you help the developer replicate the issue?
- Does that help you review your test ideas?
- Do they talk about app crash, do you have access to these logs or can you approach someone who can share the logs?
- Finally is the review a spam? Would it affect app sale? Can you remove it?

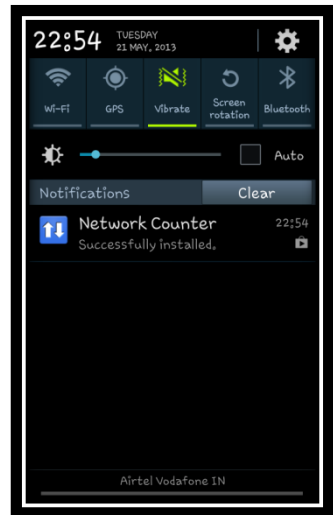
Mobile Testing: Ready Reckoner

Launch App From Notifications

When you tap on a notification, a dormant app will be launched and the UI - the notification is meant for is brought up. Sounds simple? With built-in actions within notification, there are many areas where notifications could go wrong.

Learning:

- Does tapping on a notification launch the right UI?
- What happens when you tap on an action such as 'Reply' on an email app? Does it let you compose and send or does it pull up the email UI?



Mobile Testing: Ready Reckoner



- Are notifications meant for the latest OS, backward compatible?
- Unless the app is currently active such as audio/video player, does the app really require a persistent notification on the notification bar?
- As seen in the screenshot, one can control the number of notifications too in iOS
- Also, test for the types of Alert Styles

Mobile Testing: Ready Reckoner



Splash Screen

A splash screen is the first screen you see once you tap on the app icon.

Learning:

- Does the splash screen convey any information about the app? Is the app logo, app name consistent with the App Store?
- Does it appear and disappear too quickly?
- Does the splash screen really initiate tasks in background or have just a fake timer?
- Pause/Re-launch, Kill/Re-launch – do you see splash screen?

Mobile Testing: Ready Reckoner



Landing Screen

The first screen after the Splash screen is the Landing screen for the app. Does it create a positive impression or makes you wait like the one in the screenshot?

Learning:

- If you can configure the landing screen for your app, try that test
- Landing screen must have a simple layout and be highly usable. As a user, I don't prefer several options thrown at me just after app launch.

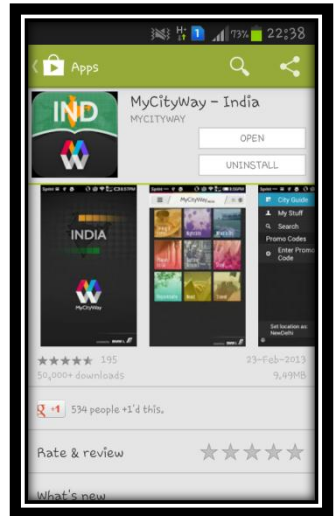
Mobile Testing: Ready Reckoner

Uninstall Using Google Play

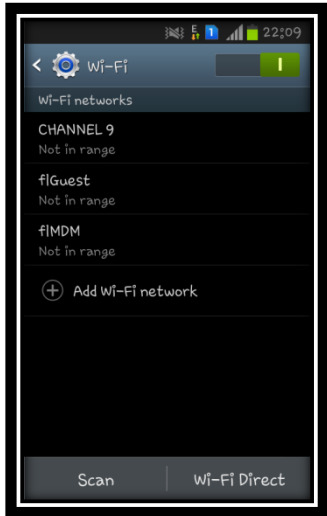
Do you install your apps from Google Play? Have you tried uninstalling apps using Google Play?

Learning:

- Once the app is installed, launch Google Play and search for the particular app. Tap on 'Uninstall' button. Is the app removed?
- Is the shortcut removed from the device pages?
- Does this process leave any residue in terms of data or images or files?
- Are there any settings reset to default settings?
- How different is this from Application Manager > Uninstall?



Mobile Testing: Ready Reckoner



Wi-Fi and VPN

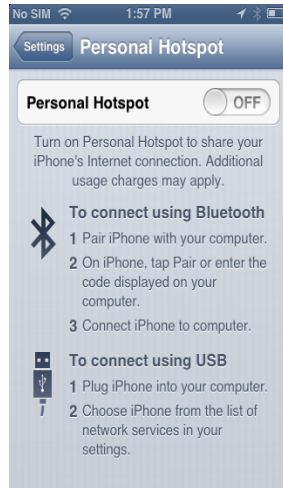
The interesting thing about Smartphones is their ability to connect to Wi-Fi, but that doesn't end there, you can even use it to share your mobile's data network.

Learning:

- In an MDM scenario when the device is restricted from connecting to a random Wi-Fi network, connecting to a Wi-Fi hotspot created on a secondary device comes in handy
- How would you ensure that the app doesn't make too many Wi-Fi network calls that could drain the battery?

Mobile Testing: Ready Reckoner

- Can the app also fall back to Wi-Fi for location in case GPS is turned OFF
- Will newer devices selectively restrict apps from using Wi-Fi and data networks?
- Connect to your corporate network via VPN and test app that requires connectivity.



Mobile Testing: Ready Reckoner

Lock Screen

A very simple yet effective test. Launch your app and lock the phone. Unlock the phone and observe the app behavior.

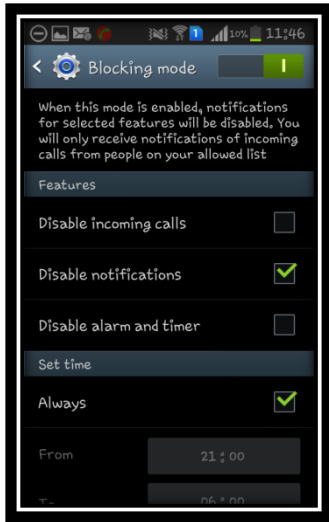
Learning:

- Does the app reset the current state and start from scratch?
- On unlocking the device, do you see app screen or home screen?
- Is there any popup for user intervention?

You can try both manual lock and auto-lock on the device



Mobile Testing: Ready Reckoner



Blocking Mode

Set the Blocking mode to On. This feature is useful to test how your app behaves when notifications are blocked.

Learning:

- App should continue to function properly
- Once notifications are enabled, pending (appropriate) notifications must be displayed
 - Existing notifications in the notification drawer should not be removed
 - Try combining the notifications feature with alarm/timer popup

Workflow: Settings > Device > Blocking mode

Mobile Testing: Ready Reckoner

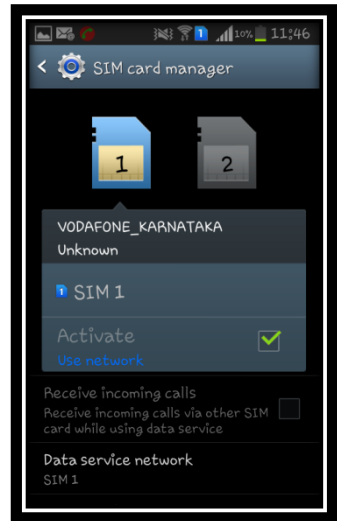
Dual SIM

If your mobile phone supports dual-sim, have both the SIM cards activated and insert them into the device.

Learning:

- If the app reads data from the SIM card, can you control which SIM card to be used?
- What happens if one or both the SIM cards are removed in between app operations?
- How about different network states of the SIM cards? What if one SIM card is set for data usage only?

Workflow: Settings > Device > SIM Card Manager



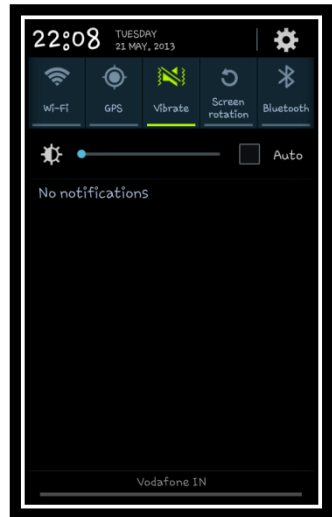
Mobile Testing: Ready Reckoner

Brightness

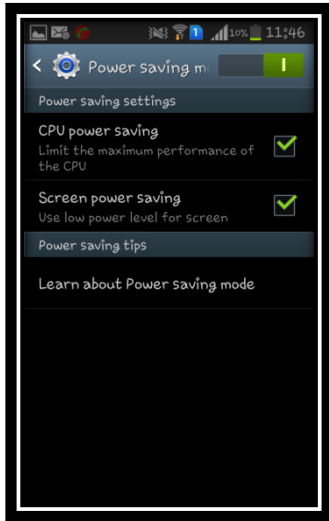
A mobile app is used in more locations than one can imagine – a coffee shop, while you are waiting for the bus, in the bus, dark meeting room or in the playground.

Learning:

- Set your device brightness to minimum, maximum and check your app display. Does any app screen color hurt the eyes
- Is it difficult to view the app screen in low or high brightness?
- Is there an option to set brightness specific to app?



Mobile Testing: Ready Reckoner



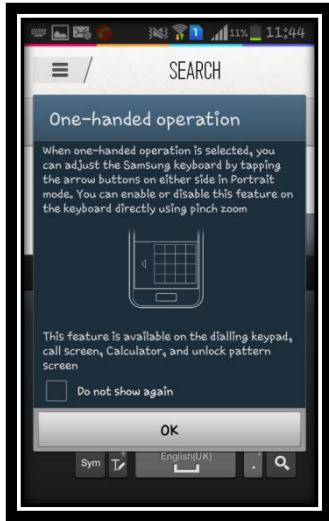
Power Saving

Have you used the CPU and Screen power saving mode?

Learning:

- Using low power for CPU means sluggish performance on the device. How does the application perform in this mode?
- Is your app UI optimized to handle low powered screen brightness?
- Does your app query these power saving modes and optimize itself accordingly?

Mobile Testing: Ready Reckoner



One-handed Operation

I found this feature by accident. Once enabled, the keypad shifts to either right or left of the screen. The user can complete most of the tasks using only one hand.

Learning:

- Using this feature, a lot of spelling mistakes were observed. The fingers used to a particular position on the keypad.
- This can be used to check an operation, its cancellation and usability of the app
- This might not be available for many devices
- Watch out for inconsistent app behavior

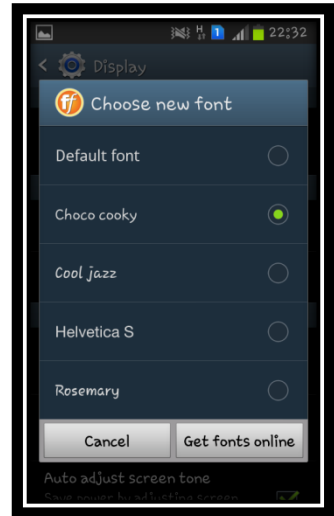
Mobile Testing: Ready Reckoner

Font

Every user is unique and has varied tastes. One such variety is the type of font one likes. With most of the smartphones giving the user an option to choose a font, users are no longer with the default font. How does your app behave when there is a font change?

Learning:

- Does the app continue to use the default font?
- Some of the symbols would not be displayed in the non-default fonts
- Change font when app is in use
- Which font is given higher priority – app font or device font?



Mobile Testing: Ready Reckoner



Font Size

Once you select the font of your choice, why not change the default font size? Some like it tiny so that they can see lot of text at one shot while others prefer Huge text so that they don't strain their eyes.

Learning:

- Which category of user groups do you belong to – The Teenager or The Grandma?
- Does text wrap itself when font size is increased?
- Does text overlap when font size is changed?
- Will the settings take effect immediately or need app restart?
- Is the font size applied consistently across app?

Mobile Testing: Ready Reckoner

Language

How many languages can you read and write? The more you know, the better especially if you are a software tester. This test idea is all about Device Language.



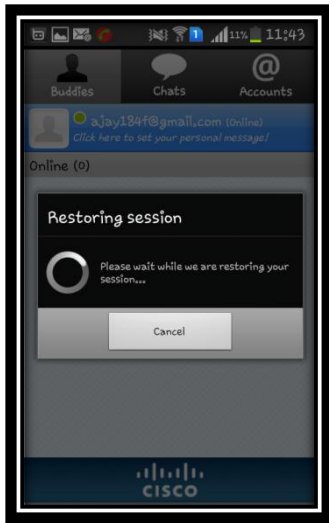
Mobile Testing: Ready Reckoner

Learning:

- Change device language from the default language to any other language.
- Change the language before launching the app. Repeat this once the app is launched. Is the app behavior affected?
- If your app has a setting to change language, which setting is given the higher priority: Device language or app setting?
- Look out for partially translated text. Also, some of the strings might be garbled.

As seen in the screenshot, there is a different setting for Language and Voice Control in iOS. Try having two different languages selected and see how the app behaves.

Mobile Testing: Ready Reckoner



Backup and Restore

Let us look at backup and restore at app context and MDM context separately

Learning:

- Backup and restore an app using a third party app. Are you able to use the app? Or at least does the app rollback to fresh install state?
- What happens when the application is backed up and restored from the cloud? Is the app data intact as expected to be?
- Does backing up application data alone and restoring have an effect on the app?

Mobile Testing: Ready Reckoner

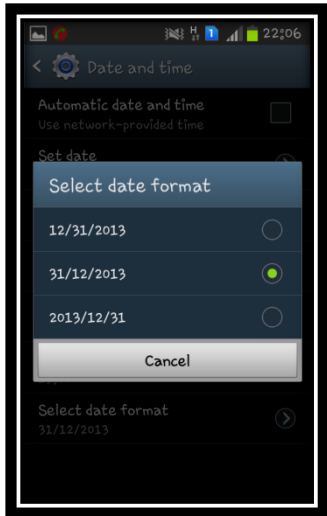


Learning from MDM context: Should apps be restricted from backup?

Mobile Testing: Ready Reckoner

- Are restricted apps that are backed up to the cloud and restored, securely wiped?
- What happens to the data that was securely wiped, if restored from the cloud?
- Are they monitored and wiped again?

Mobile Testing: Ready Reckoner



Date Format

Does your app display or takes date as input? Have you tried with different date formats?

Learning:

- As seen in the screenshot, there are at least three ways to display date. Which one is the default for your app?
- As a user, can you change the format? Does the change in system defaults affect the app setting?
- A test idea closely related is the Time Zone test idea. Test both of these together

Mobile Testing: Ready Reckoner



- On devices that have been reset to factory settings, the date and time is not set to the correct values. How does the app behave?
- Simulate the above scenario by doing a Reset Factory Settings, Install the app and then check the date format displayed

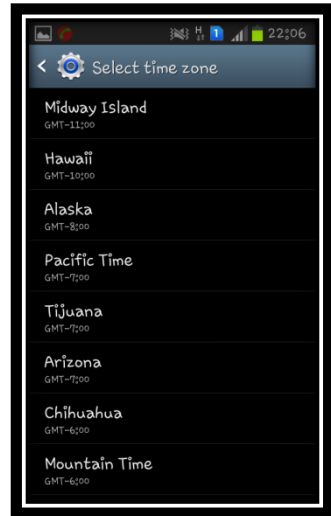
Mobile Testing: Ready Reckoner

Time Zone

How does the app behave when you travel frequently across countries? What if there are multiple time zones within the same country? Aware of the day light savings?

Learning:

- If your app is dependent on time zone a lot, check how the changes affect the behavior of the app
- Change time zone before and after launching the app
- Perform few operations after changing the time zone and reset the time zone
- As a user, can you guess what will be the

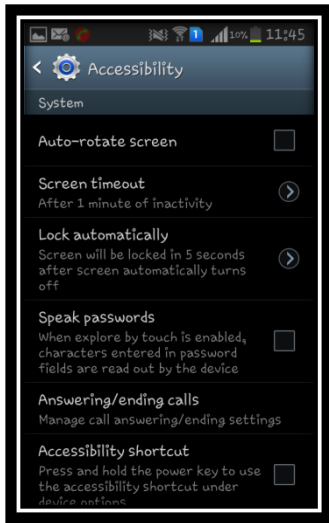


Mobile Testing: Ready Reckoner

behavior? Is the app behavior consistent and correct?

- A good test idea would be to test on multiple time zones and monitor the app for more than 24 hours.

Mobile Testing: Ready Reckoner



Accessibility

There have been companies sued for creating products not catering to Accessibility needs. Don't let your company be one of them.

Learning:

- Few Accessibility guides:

iOS: <http://bit.ly/iOSACC>

Android: <http://bit.ly/AndrACC>

Tizen: <http://bit.ly/TizeACC>

- Use device Accessibility settings and test for each of the option with respect to app screen
- Take feedback from challenged people about the app
- Some of the additional settings vary across

Mobile Testing: Ready Reckoner



platforms. Use them

- Read the Accessibility guides and check if you are violating many basic principles of Accessibility

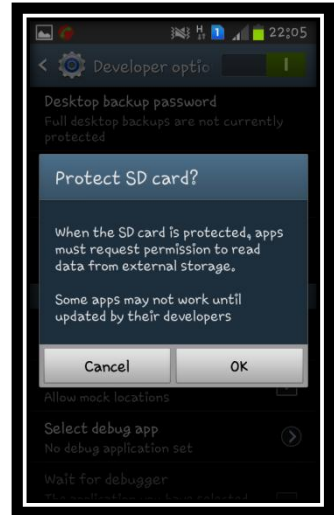
Mobile Testing: Ready Reckoner

Encrypt Device or SD Card

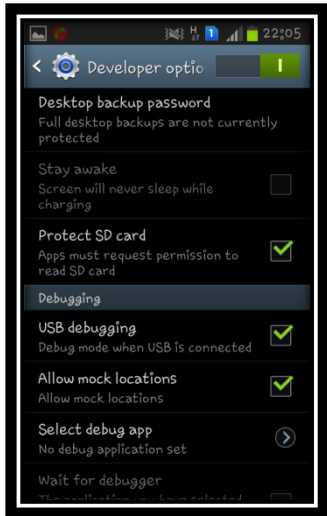
Encryption is good, alright, but have you wondered how does your app perform while accessing an encrypted SD card or for that matter on an encrypted device?

Learning:

- Does the app make best use of encryption? Did you know that apps using encrypted content can use secure video capabilities of attached displays (link: <http://bit.ly/jbprotectedcontent>)?
- Does your app prompt for permissions to access encrypted SD card content (refer screenshot)
- Compare performance ratings for your app installing on devices with and without encryption



Mobile Testing: Ready Reckoner



Fake GPS with Mock Locations

Mock locations was introduced to help developers simulate location related scenarios on the Android Emulator. Mock locations provide an easy way to test out geo-fencing related features without having to wander around. But hey, mobile applications are all about mobility, so do spend some time field testing your app.

Learning:

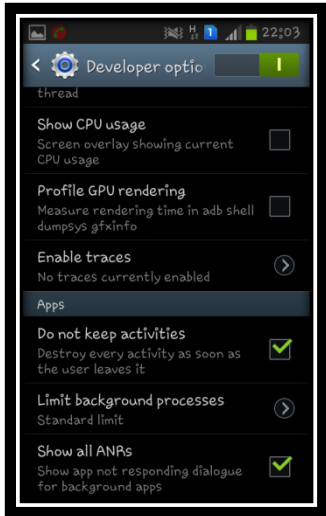
- Allow Mock Locations is available under Settings > Developer options
 - Fake GPS is the app that helps you simulate mock locations if this option is turned ON
 - Does the app pick up the simulated location?

Mobile Testing: Ready Reckoner

- Disabling Fake GPS is the best way to find if the app falls back to network or WiFi based locations
- This test idea is a good reminder that one can use third party apps to test a particular app. Scout for such apps and build your tool-kit



Mobile Testing: Ready Reckoner



Do not keep activities

The Android team was unable to replicate a CPS ticket. Customer reported that when the user opened a document on a third party viewer and pressed the back button to return to MaaS360, the app crashed.

Although we weren't able to replicate, Ajay was complaining about the same error. He was kind enough to bring his device to the Android team where we could replicate the issue. This was when Avinash asked us to check if Ajay had turned on Do Not Keep Activities under developer options.

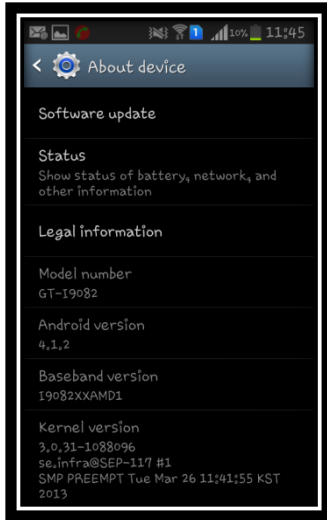
Learning:

- An Activity is the 'Window' equivalent in Android

Mobile Testing: Ready Reckoner

- When multiple UI (or Window) stack up and stay unused for a longer period, Android removes them from the memory to avoid performance issues
- When user presses the back button, the app has to retrieve the UI that was closed to show it to the user
- Do Not Keep Activities under developer options simulates this scenario to help the developer handle such instances where UI is removed by the OS but retrieved by the app
- Keep this option checked while testing an app during the QA phase

Mobile Testing: Ready Reckoner



About Device

Many testers seem to struggle in identifying details about the device. Most of the devices have a separate option presenting details about the device.

Use these screens to understand the device hardware and OS

Mobile Testing: Ready Reckoner

Logs Can Speak

IPCU or ADB, when a device is connected and watched for logs, it could speak volumes about the app.

Learning:

- Violation of privacy through personally identifiable information such as name, phone number, address, phone's IMEI, MEID, UUID shall not be logged
- Sensitive XMLs/JSONs, URLs accessed by app may not be logged. Also, app exceptions, DB locks, DB data can be traced through logs
- Passwords should not be in clear text

```
C:\Users\skrishnaswami\Downloads>adb shell
adb server is out of date. killing...
* daemon started successfully *
shell@android:/ $ ls
ls
acct
cache
config
d
data
default.prop
dev
efs
etc
fstab.herring
init
init.goldfish.rc
init.herring.rc
init.herring.usb.rc
init.rc
init.trace.rc
init.usb.rc
mnt
proc
root
sbin
sdcard
storage
sys
system
ueventd.goldfish.rc
ueventd.herring.rc
ueventd.rc
vendor
shell@android:/ $ _
```

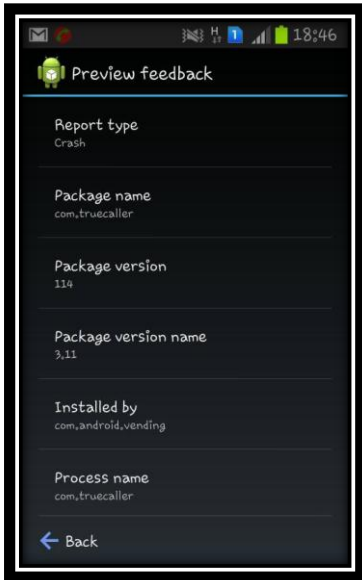

Mobile Testing: Ready Reckoner

Tools you can use:

- <http://www.satisfice.com/tools/log-watch.zip>
- iPCU in iOS
- ADB in Android
- Access built-in log generation tools on certain apps

Look for log viewers in Google Play, iTunes App store

Mobile Testing: Ready Reckoner



Crash Reports

What to do when app crashes? Click on Report and send email to the developer. If you can include your email, add it. Or else, take screenshot of each of the screens and send it to developer.

Learning:

- If you use Android Debug Bridge (ADB) on Android or iPhone Configuration Utility (iPCU) on iOS devices, it is easy to capture these logs
 - It is a good idea to always have the device hooked up to ADB/iPCU while testing and monitor the interactions
 - Over a period of time, learn to analyze logs

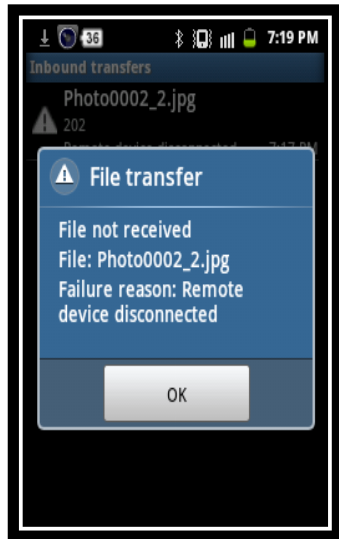
Mobile Testing: Ready Reckoner

Error Messages

Every app developer must ensure that error messages are given equal importance before app release. As software testers, it is a good idea to force the app to spit out all the error messages.

Learning:

- Remember the FAILURE Mnemonic by Ben Simo.
F: Functional
A: Appropriate
I: Impact
L: Log
U: Usability / UI



Mobile Testing: Ready Reckoner

R: Recoverability

E: Emotions

- Does the app let the user know why the error message is displayed? Is the error message meaningful?
- The attached screenshot is a good example of an error message. All the relevant details are highlighted

Mobile Testing: Ready Reckoner

Network Interruptions

There are situations when there is no network or weak signals. How does the app react in such situations?

Learning:

- Does the app let the user know immediately or the user is left to figure out?
- How does the app recover in case the connection is retrieved?
- Test with different signals – 2G, weak/strong signal, 3G, Wi-Fi



Mobile Testing: Ready Reckoner



Home Button vs Back Button

On Android application when an app is closed by pressing the device's home button, the app is left in a suspended state, whereas on a back button, the app is removed from the memory.

Learning:

- When app is closed using the home button, app should resume from where the user left
 - What is the performance impact on the device when an app is launched from a suspended state?
 - When closed using a back button the app should be removed from the memory, when re-launched database and UI states should be reinitialized

Mobile Testing: Ready Reckoner

- The home key was pressed when I was playing the below game. On re-launch, the Paused screen was displayed
- The user should be able to resume from the suspended state without any loss of data



Mobile Testing: Ready Reckoner

QWERTY Keyboard

Devices with QWERTY keyboards can be obsolete but on older devices with QWERTY keyboard, the app should perform as expected.

Learning:

- App should change orientation once the keyboard is pulled out
- When user taps on any input field, On-screen keyboard should not pop up if the QWERTY keyboard is already pulled out
- Test out apps such as financial calculator, scientific calculator, games, productivity apps which involve calculations
- Do test QWERTY keyboard on a localized app.



Mobile Testing: Ready Reckoner

Longer Operations

Have you seen this error while using an app “Application not responding’. This is the dreaded ‘not responding’ equivalent of Windows on Android or iOS caused by longer operations by the app.

Learning (plus best practice):

- The application must handle any long operation by showing a processing icon
- If larger file downloads are in progress, appropriate progress bar should be shown



Mobile Testing: Ready Reckoner



- Database encryptions or compacting files or folders can take longer time, this has to be visually indicated to the user (in the screenshot provided, the app actually encrypts docs, converts db to latest version among several other things)

Downloading resource files for images especially on games app, installing updates, downloading large documents should be avoided on a low battery.

Mobile Testing: Ready Reckoner



Resumable Downloads

Any download within the app should be resumable not only for usability but it also conserves network and battery.

Learning:

- Large downloads over 2G networks drains battery. It could trouble the user if the app does not provide an option to resume downloads from where it left
- You may also have to test by closing the app using the back button during the download, reopen it to ensure that the download continues

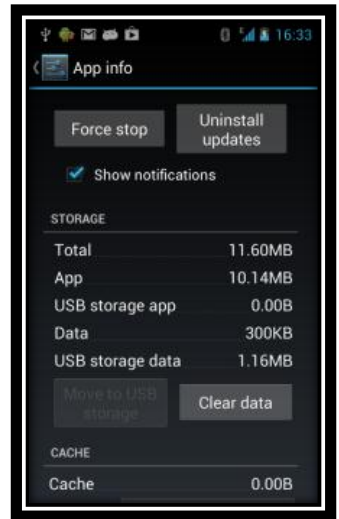
Mobile Testing: Ready Reckoner

Clear App Data

This is one of the best hacks you can perform on an Android device. Install an app and open Applications under Settings, and clear the app data.

Learning:

- Most Android applications store data on a SQLite database or XMLs. Clearing the app data could make the app unstable.
- On security apps, app data cannot be cleared most of the times
- Most of the times, the app must roll-back to a fresh install when the app data is cleared



Mobile Testing: Ready Reckoner

Kill Background Process

There are different types of apps, especially in Android, background, foreground and service apps. Let us take for example the email app on Android. Although the email app appears to be one, it spins off several 'worker apps' or service apps that takes care of specific functionalities such as email sync, downloading attachments, exchange connector etc.

Learning:

- How does the app handle when its background processes are killed?
- How does a primary app handle when its core service app could be killed and removed?



Mobile Testing: Ready Reckoner

- Does killing the primary app have any effect on the service app too?
- In iOS, you can double press the Home key and get the list of running apps. Long press on the app icon. Once the red colored – icon appears on the app icon, tap it to kill the app



Mobile Testing: Ready Reckoner

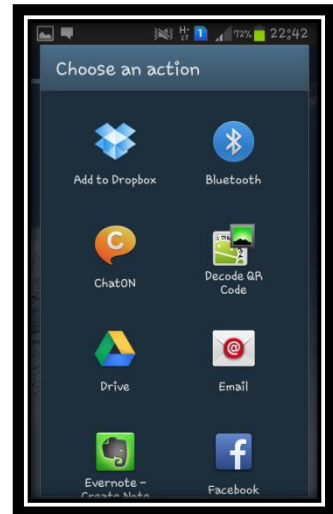
Other App Interactions

Two ways an app can interact with other apps.

- * Passing on files for a functionality like document viewer, or attachments via emails
- * The developers could agree and whitelist each other's app to perform a functionality like an MDM app can configure VPN on Cisco AnyConnect app

Learning Case 1:

- Are the third party apps prompted for a user to open, relevant to the file or operation that is about to be performed?
- Does your application gracefully exit or stay suspended when the third party app is open in



Mobile Testing: Ready Reckoner

the foreground?

- Does the third party app require the user to login? Post login does the process continue?

Learning Case 2:

- What are the security measures used or required for both apps to communicate securely?
- Do both the apps support SDK interaction? If so, what APIs can be called? Are they secure? Can any fake app call the same APIs?
- What happens when a newer version of the third party app is released?

Mobile Testing: Ready Reckoner

OS Upgrade

Trust an OS upgrade to wreak havoc on app behavior.



Mobile Testing: Ready Reckoner

Learning:

- App settings might be lost
- App data could be completely cleared
- Any operating system related setting that the app might create (such as Wi-Fi or Network connectivity settings monitored by JuiceDefender) can be reset too.

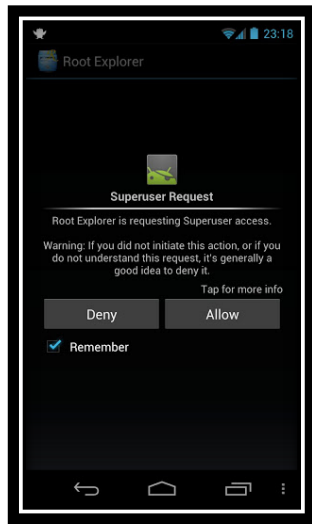
Mobile Testing: Ready Reckoner

Rooted Device

A rooted device is a potential playground for a tester. You can get access to the app's data directory, push or pull files from the app directory, remove files and also pull out the database.

Learning:

- On Android device, once you unlock the bootloader and install an 'su' binary (application), it gives super-user access to apps that desires to control certain aspects of the device
- Usually App data such as database, files, XMLs are stored under Data > Data > <App Bundle ID> directory
- You may have to provide permissions through adb shell in order to browse through these directories



Mobile Testing: Ready Reckoner

- What happens when a file that is critical to an application is deleted?
- Is the SQLite db unprotected? Does the db contain sensitive information?
- What happens when a file currently accessed by the app is deleted? Are the files encrypted?
- What happens when the files are replaced with dummies? What happens when the app is given Super User permissions?
- Does it utilize abnormal amount of resources?

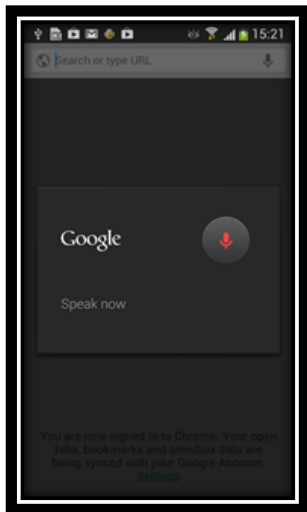
Mobile Testing: Ready Reckoner

Voice Search

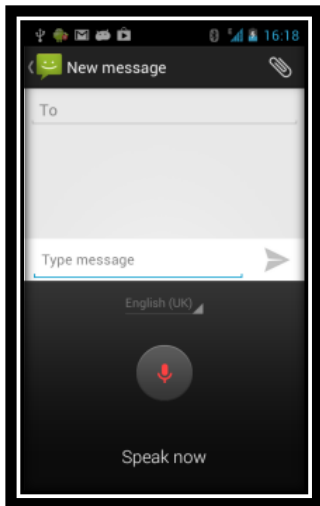
An unexpected component that you might encounter on certain app's search bar is voice search.

Learning:

- Look for accuracy of the search
- Voice search on certain apps could crash the app
- If an MDM policy such as disable microphone is enabled the voice search may not be allowed and handled inappropriately
- Try with different accents and voice pitch



Mobile Testing: Ready Reckoner



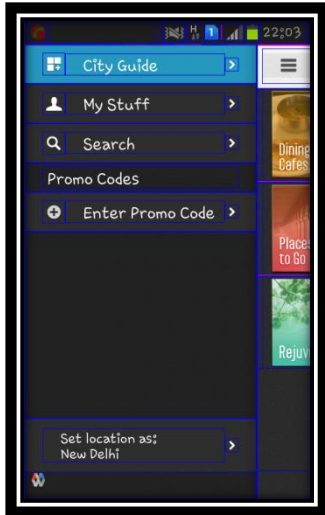
Voice Dictation

Did you know that certain keypads provided voice dictation options? Have you verified if your app supports voice dictation?

Learning:

- Validate all input fields that requires keyboard using voice dictation
- Check whether the dictation is localized or not
- If a localized keyboard is open, unsupported dictation must go away

Mobile Testing: Ready Reckoner



Show Layout Boundaries

Few devices have the option 'Show layout boundaries' under Settings > Developer options > Drawing. As seen in the screenshot, the layout boundaries help identify any alignment issues.

Learning:

- Enable the option and navigate through all the screens of the app
- Watch out for position, size and alignment of text, icons and borders
- This is a quick test to check on the device rather on the monitor or design software

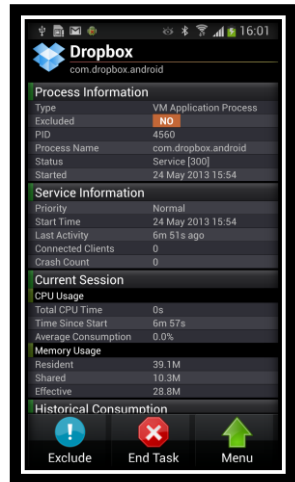
Mobile Testing: Ready Reckoner

Performance Indicators

Battery, CPU & Memory are paramount for any smartphone to operate to its optimum capacity. Apps installed can drain battery, consume more CPU or clog the memory, thereby slowing down the device.

Learning:

- Excessive logging could drain battery, consume memory
- Excessive use of network could drain battery
- Continuous UI activity, excessive wake locks on Android devices to keep the device ON, could consume more CPU



Mobile Testing: Ready Reckoner

Tools you can use:

- On iOS, Instruments, packaged with Xcode can be used to profile the app
- For Android, third party profilers such as 'Little Eye' can be used

Apps like **Battery mix**, **SystemPanelLite** (CPU/Memory), **Cool Tool** - system stats can be used (CPU/Memory/WiFi), **Caffine** - to benchmark battery usage when display is ON forever.

Mobile Testing: Ready Reckoner

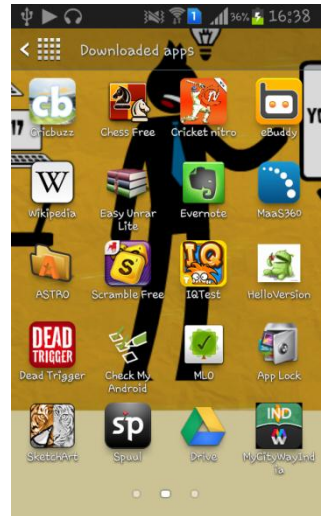
Resource Constraint

Test your app on a device loaded with apps, background processes and task managers.

Learning:

Chronic App-a-holics **Ramya** and **Abhilash** taught us that devices could be pushed to the limit with close to 100 apps on their devices. Any app installed further would crash due to limited memory and CPU. This is the best test-bed to understand app performance and behavior on devices with resource constraint.

If you notice carefully, the wallpaper of the mobile phone is also a custom one. This is a good check to test if you can recognize your app on different wallpapers.



Mobile Testing: Ready Reckoner

Tools you can use:

Install apps through iOS App Store or Cydia, Google Play or Samsung App store. Open a few apps and press Home button to continue. Install a few task or memory managers that will clean up memory when the app is used.

Mobile Testing: Ready Reckoner

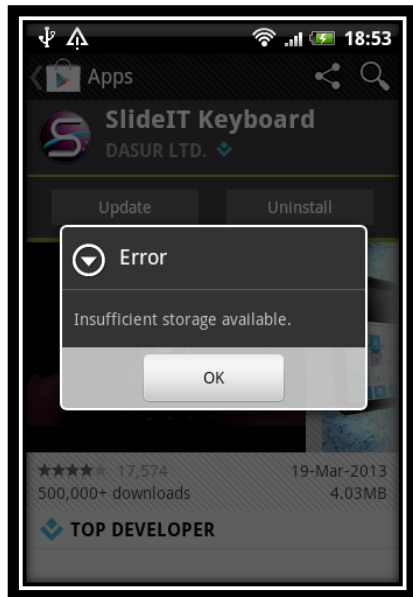
Low Disk Space

After taking a lot of pictures, video recording, the device runs out of space. Have you faced such a scenario?

Learning:

- Remove the SD card and install many apps. Run multiple apps and let them write to the device. Observe the app behavior
- Is the user notified? Does the app handle this scenario gracefully?

What happens once the disk space is increased?



Mobile Testing: Ready Reckoner

Low Battery

We have covered low battery related scenarios in other test ideas. So, let us share a few more learning here.

Learning:

- The app should avoid making network calls on a low battery scenario unless it is very critical
- Can the app destroy itself from consuming more battery and launch on battery recharge?
- Excessive logging on an app can also drain the battery, it is better if logging is switched off automatically when the app identifies low battery
- For further low battery related ideas, do refer this patent for interesting hardware related scenarios (assigned to Qualcomm)



Mobile Testing: Ready Reckoner

<http://www.google.com/patents/US20120108224>

<http://bit.ly/11l95GD>

Using quality of service to control hardware for low battery scenarios

US 20120108224 A1

ABSTRACT

Certain embodiments of the present disclosure relate to methods for reducing power consumption of a mobile device while maintaining QoS of communication links. If the available battery power is below a predefined threshold, then several power-saving techniques may be triggered that increase an air time of the mobile device by shutting down one or more modem hardware components.

Mobile Testing: Ready Reckoner

The Hidden Slider

Sliding drawers or sliders is a usability feature that partitions UI for better usability. Sliding drawers are prone to crash since two different layouts are fused together.

Learning:

- Usability, indication or a handle that shows that a sliding drawer exists, look for accessibility options
- Drawer UI when tested using monkey tester can crash the app
- Drawer behavior between Portrait and Landscape orientation
- Behavior on Phones and Tablets



Mobile Testing: Ready Reckoner

Tools you can use:

Monkey testing via adb

```
adb -d shell monkey -p <package name> -v <event count>
```


Mobile Testing: Ready Reckoner

Resources

1. Tap Into Mobile Application Testing – Jonathan Kohl
2. <http://developer.android.com/index.html> for Android
3. <http://bit.ly/AppMobileHIG> for iOS
4. http://www.cio.com.au/article/462810/10_popular_iphone_apps_-_blacklisted/
5. http://www.appqualityalliance.org/files/AQuA_top_10_app_errors_good_qa_helps_you_to_avoid.pdf
6. <https://developer.tizen.org/documentation/ux-guide/design-principles>
7. <https://sailfishos.org/develop-overview-article.html>
8. <http://msdn.microsoft.com/library/windowsphone/develop/jj247547%28v=vs.105%29.aspx>
9. <http://www.appqualityalliance.org/resources>
10. Learn about Quantum Computing
11. Keep collecting test ideas from every source

Mobile Testing: Ready Reckoner

12. Mind maps on Software Testing Club



Android Testing
Mind map



iOS Testing
Mind map

Mobile Testing: Ready Reckoner

