MOBILE EMULATORS VS. REAL DEVICES

Different aspects of app testing with emulators and real devices.

Advantages and disadvantages of Android and iOS emulator

Advantages:

- Simulates both software and hardware.
- App runs unmodified.
- Helps you find unexpected behavior.
- Tends to be free and open source. It is a fairly inexpensive solution, so the most obvious advantage is the price. All you need to do, is to download the software, install it on your PC, and you're ready to go.
- Can be connected to an IDE for early testing during development. The emulator is usually a part of the SDK provided to developers. Due to their integration with the development environment, mobile emulators provide the developer or tester with an access to detailed information, such as debugging information, which is very important for the development phase. It allows for convenient step-by-step debugging of your application on the emulator.
- Since emulators are simple client software that run locally on your PC, they run faster and with less latency than real devices connected to the local network or in the cloud.
- In addition, there may be cases where mobile emulators can give you the benefit of simulating hard-to reproduce scenarios (low battery, certain GPS coordinates, and others) that a real device connected to a network cannot easily support. However, this simulation is incomplete at best, as there is no guarantee that real devices would behave the same way in such cases.

Disadvantages:

- There are cases and situations when mobile device emulators are very slow (because they simulate both hardware and software).
- A mobile device emulator doesn’t take into consideration such factors as battery overheating, drainage, or conflicts with other (default) apps.
- Setting up a good emulator takes time and it is expensive.
- Emulators may be incompatible with the app or app elements, which means that you need to create patches here and there to keep on using the emulator.
- Emulators may support only certain OS versions.
- Even if the testing goes perfectly, you cannot be 100% sure that your data can actually apply to a real device. This raises the question of which tests need to be double-checked on a real device and which can be assumed reliable on the emulator. Also, in the event of a test that fails on the emulator, testers need to decide whether to perform the test on a real device or simply assume that the function does not work and requires correction.

- Hardware and Software Differences. The emulator is typically a light version of the OS and often does not reflect the specific hardware and software features of each supported device.
- Different Network Environment. In terms of network configuration, mobile emulators run on the PC, connect to the LAN and access the Internet via your personal/corporate firewall. Using real handsets, the network is connected to the radio interface and from there to the Internet. In many cases, the low level network stack itself is completely different. Regardless of whether this is good or bad, the network environment is different and will result in your application behaving in a different way.
- Differences in Computing Resources. As compared to mobile emulators running on PCs, real handsets have limited resources. The mobile handset is a compact platform with limited CPU power and memory. An application running on a PC does not accurately reflect the user experience on your handset, which, in extreme cases, it may not even work on.
- Mobile networks affect application behavior. Since the mobile device is still a phone, network-related events (such as incoming call, text message, and so on) must be tested to determine their impact on your application. This is very hard to accomplish using an emulator, because quality of the network varies between carriers, states, countries, and regions. Since emulators are not connected to the mobile network, they are not capable of providing an answer for how these will affect your application.

Real devices give you a real feeling of the app, how it performs on real hardware, and all factors that may contribute to making an app work or crash.

Advantages and disadvantages of real devices

Advantages

- You test in a real environment with real conditions (weather, location, network, and so on), providing real insights on the functioning of the app.
- Screen resolution and brightness can be easily tested in a series of different lighting scenarios.
- Performance of a real device is faster compared to other virtual options.
- Testing interoperability is easier.
- Easy to replicate bugs, which users have on their devices.
- No false positives, as it happens with virtual testing solutions.
- Real devices allow to test push-notification, geolocation, and the devices build-in sensors like orientation and Wi-Fi connectivity.
- Battery drainage issues derived from your app or your app in correlation with other default apps.
- You will be testing on the real OS with the manufacturer’s tweaks, not light OS versions.
- Test for incoming interrupts, like SMS and calls.
- Testing on real handsets always gives you accurate results. There is no need to worry about “false negatives” or, more importantly, the ever-dangerous “false positives.” Real handset

testing is typically performed in a live network. This is vital because network-related events could affect the quality of your application. In addition, different network technologies (such as

3G, WIFI, and others) could impact the way your application behaves.

- Testing on real devices is the only way to truly understand the user experience, taking into account the CPU, memory, screen size, etc. for a given device. When it comes to the quality of your services, there is no substitute for using a real handset operating in one or more live carrier networks.

- Furthermore, it is much easier to expose performance defects with real handsets, as well as defects which are the result of the handset itself or its environment. These types of defects cannot normally be detected using emulators.

Disadvantages

- The real devices are costly compared to the emulators. Thereby projects under budget and timeline constraints may risk profitability as well as the viability of the overall project.
- There is a wide range of mobile devices from Apple to Samsung, from Android to Symbian, and so on. Considering this variety 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.
- When real mobile devices are used in the developing stage for unit testing and similar purposes, it could turn out to be harder to connect to the IDE than the emulators and it causes big problems for debugging. In a project with timeline constraints, this may hamper the overall conclusion of the project.
- 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 process 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 URLs. The adoption of these procedures would ensure that a lot of important memory space may be eaten up thus impacting on the overall performance of the application.

Read more: **Mobile Emulators vs. Real Devices**