What is Interface Testing?

Interface Testing is a kind of software testing which verifies the communication between two different software systems. It checks the authentication of the connection established. Interface Testing includes testing of two main segments

- **Web server and application server interface**
- **Application server and Database server interface**

Why do we need Interface Testing?

To ensure that the end-users or customers should not encounter any problem while using a particular software.

To identify which application areas are usually accessed by end-users and to check if they are user-friendly.

To verify the security requirements while communication propagates between the systems

To check if a solution is capable of handling network failures between an application server and website

To make sure that the software is proficient in error handling.

Approach for Interface Testing

Define your requirements properly

Start building small codes and then testing them, instead of directly jumping on making large pieces of codes

Find some automation tools

Mark the start and the stop points, that will eventually help you in checking the performance of the test

Safeguard Your Business With UI/UX Testing

We live in an ultra-competitive market, businesses should focus on creating a top-notch Graphical User Interface to attract users. Don’t allow the same bug to bite you twice. Deliver mobile apps with confidence with proactive Mobile User Interface Testing Services.

Let’s face it; if your mobile app has many bugs and constantly glitches out, your end-users will probably bash it and trash it. Mobile user interface testing services or GUI (Graphical User Interface) testing services are specifically designed and implemented to detect and weed out all the bugs in an app before its official release to deliver a smoother experience to your end-users.

User Interface testing services assess the functionality, performance, and effectiveness of an app’s user interface. Thoroughly check every element of your mobile app to make sure it has a user interface so copacetic that it delivers a flawless experience so that you get the best out of your efforts.

To Ensure Consistent Appearance, Accuracy, Logical Flow, And Consistency. Both automated and manual user interface testing methods are used concurrently to thoroughly test the user interface of an app, improve app quality, and significantly increase its usability.

Automated UI/UX Testing

To quickly test your app’s performance under different conditions, automated coded UI tests are developed and deployed by the developers using advanced User Interface testing tools like Selenium Grid and Jasper JS.

Manual UI/UX Testing

Team of expert manual testers will check every button, form, and element of your mobile app and will sniff out the most difficult of interface bugs.

Smart Interface Testing Process

Unit Testing

At this level, the basic code of your mobile app is tested. To make sure that the principle of your app works correctly by testing its particular elements, mainly classes, functions, and interfaces.

Integration Testing

Mobile applications mostly misbehave when certain specific units are integrated. At this level, need to test the behavior and functionality of your app after each unit integration.

System Testing

In System Testing, all the components of your app are tested as a whole to ensure that it passes all the preferred design standards and conforms to the specified requirements.

Acceptance Testing

At this stage, to make sure that your app complies with the specified user requirements and is ready for the final release.

Among the variety of types, applied to test the mobile app at different stages of its creation, we should definitely mention the following list.

Usability testing. It is carried out from the early stages of app creation to verify if the app fulfills the established objectives and tasks getting a good response from users. The primary focus of this testing is easy and quick use of an app, simple onboarding and user's satisfaction with the entire experience. For higher efficiency and productivity of general creative flow, this type of testing should start much earlier than any single line of code will be written, from the first schemes and transitions put into UX wireframes.

Installation testing. At the initial stage of installation, the app should add required software to the device automatically. And uninstalling, it should remove all the available content and databases from the device which are used by the app. So, this type of testing checks that the install/uninstall flow goes properly.

Functional testing. It is the most basic test for any app to ensure that it is working according to the defined requirements and there are no functions missed in the process of interaction. Functional testing mainly includes finding possible specific bugs of the device or navigation issues of the app. This type of testing should be done at the primary stages of development. It enables developers or testers to check and measure database or network queries with response times, crashes and memory issues.

Performance testing. Rather a stressful part of any app testing is performance test revealing the omissions which left unnoticed during functional and user interface testing. This testing is required to be done on the actual device only so it means the whole app is coded t this stage. This type covers client-side, server and network performance. For example, it checks the performance specifications and behavior of the app under certain conditions such as low battery, bad network coverage, low available memory and etc.

Interruption testing. An app may face various interruptions being in active mode, such as incoming calls or network coverage outage and recovery. This kind of testing shows what the app will do in this variety of cases. The common types of interruptions are:

Incoming and Outgoing calls, SMS or MMS and different notifications
Low memory warning
Cable insertion or removal
Network outage or recovery
Media Player on/off
The device power cycle, such as low battery notification.

Memory testing. This type checks that each app maintains optimized memory usage throughout surfing. As mobile devices have definite limits of available memory, this testing is crucial for the appropriate product functioning.

Security testing. It checks the vulnerability of the app to hacking, authentication and authorization policies, data security, session management and other security standards. The fundamental objective of security testing is to ensure that the app data and networking security requirements are met according to standards.