BOTm

End-to-End App Testing – A 360° Approach

INDEX

INTRODUCTION

The software industry has increasingly brought the world virtually to the user's fingertips! Sitting at one corner of this great planet, people can effectively reach across the continents to communicate, shop, pay, play, pray, educate, medicate, entertain, and the list goes on... The user's interaction with an app does not stop at one activity per app. The web and mobile app world must necessarily permit a host of permutations and combinations from this endless list of activities, in order to provide users with superior viewing experiences. Consequently, a single app has to depend on various sub-systems to give users what they need.

This technology-driven combination of sub-systems can get quite complex, since the various components of the app may not be from the same source system, and furthermore it frequently involves systems of other enterprises. A shopping site will in addition to its own complex web of internal systems, have to integrate external sub-systems of the sellers, banking channels, etc. As the number of sub-systems – both internal and external increase, the app's complexity also increases, needless to say, testing too becomes more complex.

It is pertinent to remember that an app is as strong as its weakest link! Hence for the app's success, it is vital, that every subsystem – internal or external, and the way each sub-system links up to the others and to the main system – must be thoroughly checked to ensure smooth flow of processes for the end user. Here's where End-to-End Testing steps in to play that vital role which gives stakeholders confidence in the app, and provides users with smooth error-free operations, thus contributing to the app's success.

Given the importance of E2E Testing in the contemporary and future app industry; and the explosive number of mobile apps world-wide, this Whitepaper seeks to present a comprehensive view of End-to-End Testing with a focus on Mobile Apps.

ABSTRACT

End-to-End (E2E) Testing is the methodology used for testing an app for all layers, from the point of view of users. Hence it starts with the front-end, and tests the flow of all processes right up to the back-end, including the app's interfaces, and final endpoints. E2E Testing lays emphasis on the end user's experiences and is therefore executed in real world scenarios which may be simulated. The reason why E2E Testing is gaining popularity is because it tests the entire flow of the app; increases test coverage by testing the app's internal and external sub-systems; helps timely identification of performance issues; and greatly contributes to enhancing the app's productivity. Thus E2E Testing generates greater confidence in the app, and this is the reason why it is being widely adopted in the software industry.

This Whitepaper titled 'End-to-End App Testing – A 360° Approach', covers the various aspects of E2E Testing and has been divided into two sections.

The first section titled 'The What, When, and Why of End-to-End (E2E) Testing', is an overview of E2E Testing. This section will provide insight into what E2E Testing is all about, and also distinguish it from other related types of app testing. Additionally the 'When' and 'Why' aspects of E2E Testing will also be explored.

The second section titled 'The How of End-to-End Mobile App Testing', will focus on E2E Testing of mobile apps. This section will provide pointers for putting in place a good E2E Mobile App Testing Strategy. It will also touch upon the methods of E2E Testing which are basically – Horizontal E2E Testing, and Vertical E2E Testing.

It is hoped that this Whitepaper will provide the reader with good insights into E2E Testing.

THE 'WHAT, WHEN AND WHY' OF END-TO-END (E2E) TESTING

What is End-to-End (E2E) Testing?

E2E Testing is a methodology used for ensuring that apps perform as desired and that the flow of data is sustained for every kind of user task and process. This type of testing technique starts from the end user's perspective and simulates the real-world. Thus E2E Testing tests the software from start to finish verifying that the entire app functions as expected. It also checks the app's dependencies on other systems, making sure that all components integrate well and that process-flows work as anticipated. In short, it involves validating the app from a user's viewpoint, verifying the integrity of its data, and ensuring that integrations are syncing well together. Examples of E2E Testing: Validating the login function or the loading of a web page; verifying email notifications, online payments, etc.

For a clearer perception, it would help to understand E2E Testing in conjunction with other types of related testing like Unit Testing, Integration Testing, Functional Testing, and System Testing.

As the name suggests, Unit Testing involves testing of each minute unit or module of an app e.g. each API endpoint, or each function. Integration Testing takes over from there, to validate that each tested module of the app then integrates well with the other modules, ensuring that there are no glitches when they combine. Functional Testing aims at ensuring that the tested software meets the business requirements of the app, viewing it from a single user's perspective; endorsing the results of each test for inputs and outputs. System Testing tests all the components of the app, to verify that all functional and non-functional aspects are met.

However, E2E Testing focuses on the actual flow of a system to ensure that the functional requirements are achieved. Thus, E2E Testing differs from the others in so far as it focuses on user perspectives, and checks the app's performance across the various apps and user groups, to ensure that processes flow smoothly after changes are made, and it also seeks to validate the completion of every step in the process.

When is E2E Testing Required?

End to End testing is done on the final software app and systems. The testing chain which starts with Unit Testing; moves on to Integration Testing; Functional Testing; then on to System Testing; after which E2E Testing is conducted. Thus E2E Testing becomes the review of the completed system. However, in case any output does not match the expected criteria, or in case any other issues are encountered, E2E tests will need to be re-run. In such cases, the findings should be recorded and analyzed, with the focus being on pinpointing the source of the problem, which should of course be rectified, before conducting a re-test.

Why E2E Testing?

The primary goal of E2E Testing is to test the app from the end user's perspective by simulating real users' scenarios. Superior viewer experience is very vital in the app world, and hence E2E Testing plays an important role in the app's success, as it validates the software system and its various modules from the perspective of groups of users. E2E Testing verifies the smooth and uninterrupted flow between the various sub-systems of the app, and thus averts risks of app failure in the hands of the end user.

As mentioned earlier, apps today comprise of many sub-systems and even a small glitch in any one of the sub-systems can spell the failure of the app. This is why E2E Testing is vital. It brings with it the benefits of increased test coverage; detecting bugs before users encounter them; and reduces both costs, as well as 'go-to-market' time.

Some of the technical reasons for conducting E2E Testing are as follows:

- To ensure the fitness of the app by verifying its back-end or database layer which is central to the app's proper functioning.
- To detect errors in diverse environments like cloud, SOA-based environment, or distributed testing environments.
- To confirm the apps proper functioning over multi-tiered architecture and system workflows so that the response over connected systems can be verified.
- To provide superior, uninterrupted user experience by testing the app's front-end and user Interface (UI) aspects across platforms and environments.
- To perform repetitive tests for various processes and for diverse operations that take place in the app's functioning.
- To validate the system and its components or modules for integration and data integrity.
- To validate the app in its entirety and confirm the veracity of sub-system flows.
- To enhance test coverage and generate confidence in the app.

With this background on the What, When, and Why of E2E Testing, this Whitepaper will move on to explore how E2E Testing is done.

THE 'HOW' OF END-TO-END MOBILE APP TESTING

This section strives to answer questions related to the 'How' of E2E App Testing, with a focus on Mobile Apps. For any testing to be successful, the right strategy needs to be put in place, and hence this is what will be expounded below.

E2E Mobile App Testing Strategy

01 Creation of an E2E Test Plan

Apart from the normal test plan that covers initial mobile app testing and Regression Testing; an E2E Mobile App Test Plan, needs to be created. This should incorporate a set of essential criteria to be tested, and also contain testing strategies to verify that user requirements are met. The criteria should include app features, components, sequence, timing, and data conditions. Additionally the different test cases must be described in the wireframe so as to provide the base guidelines for the other steps in the testing process.

02 Introduce an Automated E2E Testing Framework

E2E Testing seeks to simulate the real world and ensure the app's success from a user's perspective. Hence to incorporate accuracy into the testing process, it is advisable to automate the E2E Testing framework, since it provides the guidance, tools, practices, etc. for creating and designing test cases based on user requirements. By co-relating requirements and defects, automated frameworks greatly aid the process of detecting and fixing bugs; promote test reusability, and quicken test creation and maintenance. It is important to evaluate which framework best suits the app's requirements, in order to maximize efficiency, accuracy, test coverage, cost-benefit ratio, and return on investment.

03 Devise a Good Quality Assurance (QA) Plan

User satisfaction is vital for the success of mobile apps, as there are literally millions of apps vying for user attention. Hence it is absolutely essential to have in place a well-thought QA strategy, to help promote superior user experience consistently. The QA plan must cover cross-platform functionality, performance, security, and usability issues. To achieve success in QA, the testing criteria needs to be set; testing integrations implemented; adequate time must be allocated for each process; bug fixing must be prioritized based on the importance and frequency of the affected features; QA budgets need to be defined; and QA pursuance should be a prime focus throughout the software development life cycle (SDLC).

04 Ensure Cross-browser, Cross Device, Cross-environment Testing of the App

User preferences vary across devices, browsers, operating systems (OS), screen reading tools etc. Furthermore user environments differ as they may be located in various parts of the world with varying network providers, speed levels, connectivity etc. Since E2E Testing focuses on user experience, it needs to necessarily cover all popular device-browser-OS combinations that the app's users may choose, and additionally cover combinations that targeted users may potentially opt for. To achieve success in cross browser testing, first run the design and functionality tests on the browser in use, to gauge the performance, look, and feel of the app. Then decide what is to be tested and determine the browser-OS combinations expected to be frequented by the app's users. Next choose whether to test on emulators, simulators, or virtual machines, and install the relevant browser/s for testing on the chosen infrastructure. Cloud-based testing can be considered for improving testing economy. The tests can then be executed; results shared; and problems addressed, for ensuring successful cross-browser testing.

05 Fix All Bugs and Glitches Before the App's Go-live

Bugs and glitches detected in the app's functionality should be addressed and fixed at the earliest and as far as is possible, before releasing to production. Issues related to app speed, the app's responsiveness, user friendliness, etc. must be thoroughly tested and fixed as these issues often result in users discarding the app. On detecting the bug, its nature and impact on other codes and on the app as a whole, needs to be determined and sent to the relevant team/s for fixing. Post fixing, the affected code/s and integrations need to be retested and certified before the app's 'go-live' time.

06 Confirm Proper Functioning of External Dependencies

Apps being made up of various internal and external modules require multiple integrations through files, libraries, packages, or plugins, for the various operations to flow smoothly. It is therefore important to ensure that all these external dependencies, function as they should. This dependence on other packages and libraries can be a cause of concern as they are based on external codes. Here are some pointers to ensure that the external dependencies are performing well. Ensure correct use of third-party databases and external web services; put in place libraries for gauging app adoption; always use the updated versions of third party libraries; ensure correct implementation of functionalities; fix performance problems arising from bugs in external codes; verify unknown malicious code.

Some of the technical reasons for conducting E2E Testing are as follows:

Horizontal E2E Testing

This is a widely used method especially in ERP (Enterprise Resource Planning) Apps where testing is done horizontally across the various apps that go into the making of the main ERP App. A case in point is an online Shopping Portal that will combine Inventory Systems, Accounting Systems, Shipping Modules, etc… Horizontal E2E Testing can conveniently be executed across these systems.

Vertical E2E Testing

Here E2E Testing is conducted sequentially with each module or component being tested from start to end. Testing is done in layers and executed in a hierarchical order. Vertical E2E Testing is more suited for testing vital modules or components of complex computing systems that largely do not involve users or interfaces.

CONCLUSION

The complexity in the app world has greatly increased with apps having to depend, not just on internal modules that are not from a single source, but also on external modules or components that relate to third party enterprises. This multiple module integration makes the structure of modern apps quite complex. With increased external dependencies, comes increased testing complexities, requiring multifaceted error detection and fixing techniques.

In this scenario, in order to address the testing complexities, it is necessary to adopt End-to-End Testing to generate greater confidence that all modules are working as expected, and that all integrations and process-flows are smooth and error free. With E2E Testing, the app is tested across different systems, devices, layers, servers, database interactions, APIs, and communications, which is vital for providing users with superior viewing experiences. In general, it is better to automate E2E Testing as it provides greater comfort level by increasing accuracy, testing speed, productivity, and efficiency; and making testing more cost effective.

For mobile apps, BOTm is a good centralized option for End-to-End Testing, as it is an automated platform that incorporates the latest in technology, and also covers the entire spectrum of mobile app testing. Visit www.botmtesting.com and sign up for a Free Trial.

GET IN TOUCH

☎ 022 4050 8200

✉ sales@botmtesting.com | www.botmtesting.com

BOTm is the accelerator BOT for automated and manual testing ofmobile applications - developed for both Android and iOS devices.