**Functional Testing of Mobile Application**

The Mobile Application Functional Testing is a process of testing the functionalities of mobile applications, such as user interactions, as well as testing the transactions that users may execute. The main aim of functional testing of mobile applications is to ensure consistency, fulfil the requirements defined, the risk of errors and satisfy the consumer. The various factors which are relevant in functional testing are:

• Application form based on uses of business features (bank, gaming, social or commercial)

• Target category of audience (consumer, business, education);

• Distribution channel (e.g. Apple App Store, Google Play, Direct Distribution) used to spread the application. The most fundamental test scenarios in the functional testing can be considered as:

1. To verify whether all necessary mandatory fields act as needed 1. To verify whether all necessary mandatory fields act as needed.
2. To verify that the compulsory fields are represented in a distinctive way on the computer than the non-compulsory fields.
3. To verify whether the application functions as needed when the user begins / stops
4. To check that the programme enters minimised mode if an incoming phone call occurs. We must use a cell phone, to call the device, to verify the same.
5. Validating that the phone can store, process and receive SMS while the device runs. To verify the same, we need to use a separate device to call sms to the system being checked and where the program is currently running under test.
6. To verify that the system is capable of fulfilling needed multitasking requirements whenever possible.
7. To verify that the framework provides appropriate choices for the social media network like posting, sharing and navigating etc.
8. To verify that the require regular any online payment transaction, as required by the application, such as Visa, Mastercard, PayPal etc.
9. To validate as required the page scrolling scenarios are allowed in the application.
10. To verify that the application's functionality among relevant modules is as per the specification.

To check that the truncation failures are at an acceptable maximum absolute.

1. To confirm the user receives a correct error message such as "Network error." For some time, please try "if there is a system error.
2. To verify that the enabled programme helps other applications to work satisfactorily, and does not steal in the other applications' memories.
3. To verify that when a hard reboot or device crash occurs the programme resumes at the last service.
4. To check that the programme can be implemented smoothly, given the user has the required resources and this does not result in any major errors.
5. To verify that the developers to identify auto-start facility as needed.
6. To check that the programme operates 2 g , 3 g and 4 g in all versions of Mobile according to the requirement.
7. Regression Testing to discover new software glitches in existing areas of a system after improvements have been made. Often rerun previously performed experiments to assess that the conduct of the system has not changed due to the changes.
8. To check if the application offers a reference manual for those who are not aware of the system

**Performance Testing Test Cases**

The basic purpose of this type of testing is to ensure that the programme works acceptably under certain performance conditions, such as access by a large number of users or elimination of a crucial part of infrastructure such as a database server.

The general efficiency evaluation scenarios in a Mobile approach are as follows:

1. Determining if the application is operating under specific load conditions as per the requirement.
2. Evaluate whether existing network availability is capable of serving the program at peak, median and minimum user level.
3. Determining if the current configuration client-server setup offers the optimum level of performance needed.
4. Identify the different bottlenecks in application and infrastructure that prevent the software from performing at the appropriate acceptability rates.
5. To check whether the app's system response is according to the specifications.
6. To test the software and/or equipment to assess if it is capable of handling the expected volumes.
7. Describe the various application and infrastructure bottlenecks which prohibit the software from executing at the correct acceptability levels.
8. To verify if the response time of the app is as per the requirements.
9. Testing the software and/or facilities to determine if they are capable of handling the volumes predicted.
10. To verify durability of the programme if there is a stringent user load.
11. To verify output of the network when moving around with the computer.
12. To validate the output of an application when only intermittent connexion phases are necessary.

**Security Testing Test Cases**

The basic goal of safety test is to validate that the system and information protection specifications of the system are met according to the guidelines.

The following are the most important places for Mobile device security reviews.

1. Authenticate that the software can handle any brute - force attacks that is an automatic trial and error mechanism used to guess the username , password or bank account numbers of a person.
2. To check that an application does not allow an intruder to access sensitive content or features without successful verification.
3. To validate that the application has a strong password protection system and it does not permit an attacker to obtain, change or recover another user’s password.
4. Validating that the implementation is not suffering from insufficient expiry of the session.
5. Identify the complex dependencies and take steps to prevent any intruder from accessing them.
6. For stopping attacks linked to SQL injection.
7. Identifying and recuperating from any unmanaged computer scenarios.
8. In ensure that the certificates are checked, the programme does or does not enforce Pinning Certificates.
9. To defend the device and the network against denial - of - service.
10. To examine the criteria for processing data and validating data.
11. . Allowing access control to prevent unauthorised users from accessing unsolicited information.
12. Checking if any cryptography software is broken and making sure it is patched.
13. To check that the implementation of the business logic is stable and not prone to any outside threat.
14. Determine some limitation and fix certain problems to evaluate file system interactions.
15. For example, to validate protocol handlers attempting to reconfigure the app's default homepage using a malicious iframe.
16. To guard against injections by the malicious customer side.
17. To guard against harmful injection during runtime.
18. To investigate the encryption of files, and to avoid any malicious possibilities.
19. To avoid insecure storing of data inside the applications' keyboard cache.
20. To examine cookies and avoid cookies from performing any malicious acts.
21. Twenty. Providing periodic audits to assess data security.
22. Investigate design-created files and protect the custom-created files from malicious behaviour.
23. To avoid sql injection and cases of memory leakage.
24. Analyzing various data streams and avoiding any security flaws from them.

**Usability Testing Test Cases**

The Mobile application's usability testing process is conducted to provide a simple and easy phase application with less functionality than a slow and demanding application with more features. The key goal is to make sure that we end up with a user-friendly, intuitive and identical frameworks that are commonly used by industry.

1. To ensure the controls are of the appropriate size and are suitable for broad fingers.
2. To ensure the buttons are positioned in the same screen section to prevent confusion for end users.
3. To ensure the icons are normal and compatible with the query.
4. And make sure the buttons that have the same feature will also have the same colour.
5. To ensure the validity of the zoom-in and zoom-out facilities for tapping should be allowed.
6. To ensure that the input of the keyboard can be sufficiently minimised.
7. To ensure that the user offers within an appropriate time a way to go back or reverse an action, to touch the wrong object.
8. To ensure the dynamic menus are not overwhelmed because they must be used quickly.
9. To keep the text easy and transparent, to make it available to the users.
10. To ensure the end users are accessible for the short sentences and paragraphs.
11. And make sure the font size is wide enough to be readable, and not too big or too small.
12. Validating the application requires the user to download vast quantities of data each time the user begins downloading which might not be beneficial to the success of the application.
13. To check that the program is being closed from separate states and to verify whether it is re-opening in the same state.
14. Ensure all string are translated into correct languages if there is a language translation facility.

15. To ensure that the software items are synchronised as per user actions at all times.

16. To ensure that the end customer is provided with a user manual that allows the end user to know and run the software, which may not be familiar with the procedures of the application