**CHAPTER 10: TESTING**

**10.1 TESTING PLAN**

As we have developed an Android Project, we have used Compatible Test Suite (CTS), an inbuilt testing and verifying tool for android applications inside Android Studio 2.0 itself.

1. **2 LEARN MORE ABOUT CTS**
   * 1. **How does it work?**

The Compatibility Test Suite (CTS) is a free, commercial-grade test suite, available for download as update in Android Studio. The CTS represents the "mechanism" of compatibility.

The CTS runs on a desktop machine and executes test cases directly on attached devices or an emulator. The CTS is a set of unit tests designed to be integrated into the daily workflow (such as via a continuous build system) of the engineers building a device. Its intent is to reveal incompatibilities early on, and ensure that the software remains compatible throughout the development process.

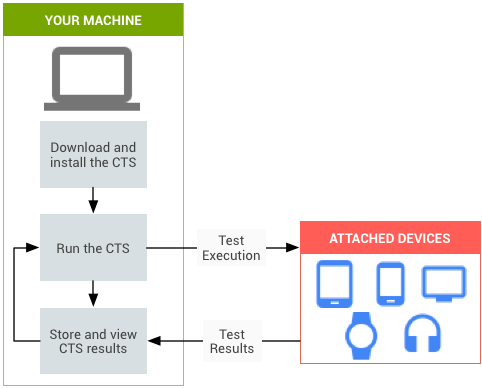
The CTS is an automated testing harness that includes two major software components:

* The CTS tradefed test harness runs on your desktop machine and manages test execution.
* Individual test cases are executed on the Device Under Test (DUT). The test cases are written in Java as JUnit tests and packaged as Android .apk files to run on the actual device target.

The Compatibility Test Suite Verifier (CTS Verifier) is a supplement to the CTS. CTS Verifier provides tests for APIs and functions that cannot be tested on a stationary device without manual input (e.g. audio quality, accelerometer, etc).

The CTS Verifier is a tool for manual testing and includes the following software components:

* The CTS verifier app that is executed on the DUT and collects the results.
* The executable(s) or script(s) that are executed on the desktop machine to provide data or additional control for some test cases in the CTS Verifier app.
  + 1. **Workflow**



* + 1. **Types of test cases**

The CTS includes the following types of test cases:

* ***Unit tests*** test atomic units of code within the Android platform; e.g. a single class, such as java.util.HashMap.
* ***Functional tests*** test a combination of APIs together in a higher-level use-case.

Future versions of the CTS will include the following types of test cases:

* ***Robustness tests*** test the durability of the system under stress.
* ***Performance tests*** test the performance of the system against defined benchmarks, for example rendering frames per second.
  + 1. **Areas covered**

The unit test cases cover the following areas to ensure compatibility:

|  |  |
| --- | --- |
| Area | Description |
| Signature tests | For each Android release, there are XML files describing the signatures of all public APIs contained in the release. The CTS contains a utility to check those API signatures against the APIs available on the device. The results from signature checking are recorded in the test result XML file. |
| Platform API Tests | Test the platform (core libraries and Android Application Framework) APIs as documented in the SDK Class Index to ensure API correctness, including correct class, attribute and method signatures, correct method behavior, and negative tests to ensure expected behavior for incorrect parameter handling. |
| Dalvik Tests | The tests focus on testing the Dalvik Executable Format. |
| Platform Data Model | The CTS tests the core platform data model as exposed to application developers through content providers, as documented in the SDK android.provider package: contacts, browser, settings, etc. |
| Platform Intents | The CTS tests the core platform intents, as documented in the SDK Available Intents. |
| Platform Permissions | The CTS tests the core platform permissions, as documented in the SDK Available Permissions. |
| Platform Resources | The CTS tests for correct handling of the core platform resource types, as documented in the SDK Available Resource Types. This includes tests for: simple values, drawables, nine-patch, animations, layouts, styles and themes, and loading alternate resources. |