Chapter 5

Software Testing

This chapter includes some test cases for the game to check if the game works properly in various situations. We are giving four test examples for four different situations here

5.1 Testing Approaches

There are broadly two types of testing approaches:

5.1.1 Black-Box Testing

It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'OK', and problematic. In this testing method, the design and structure of the code are not known to the tester.

The checklist is split into five different fields:

- Device specific characteristics. These are characteristics that are related to the device on which the app is installed.
- Network specific checks
- App checks. These are things to check that have to do with functionality that is frequently used in an app.
- App User interface checks.
- Store specific checks.

Program Specific Checks

#	Description	YES/ NO?	Remarks
1.1	Can the program be installed on the device?	Yes	Working fine
1.2	Does the program behave as designed/desired if there are different sizes of pdfs?	Yes	Working fine
1.3	Does the program behave as designed/desired if there is different forms of resumes?	Yes	Working fine

1.4	Does the app behave as designed/desired if the network is slow?	Yes	Working fine
1.5	Does the app behave as designed/desired if the network is very slow?	Yes	Working fine
1.6	Does the program behave as designed/desired if the program is used on mobile?	Yes	Working fine
1.7	Does the UI interact with the GPS sensor correctly (switch on/off, retrieve GPS data)?	No	No gps location involved
1.8	Is the functionality of all the buttons or keys on the UI defined?	Yes	Working fine
1.9	Verify that buttons or keys which have no defined function have no unexpected behaviour on the app when activating.	Yes	No unexpected behaviour
1.10	In case there's a true "back" button available on the UI does the "back" button take the user to the previous screen?	NO	You can use the browser back button
1.11	In case there's a true "menu" button available on the device, does the menu button show the program's menu?	Yes	It will
1.12	In case there's a true "home" button available on the program, does the home button get the user back to the home screen of the device?	Yes	It will
1.13	In case there's a true "search" button available on the device, does this get the user to some form of search within the resume?	NO	It will not
1.14	Does the program behave as designed/desired if the "parameters are not found " message is pushed by parser	Yes	Working fine
1.15	Does the program behave as designed/desired if some columns are found missing by the parser?	Yes	Working fine

Table 5.1: Black-Box Testing Test Cases

5.1.2 White-Box Testing

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as 'Structural' testing. In this testing method, the design and structure of the code are known to the tester. Programmers of the code conduct this test on the code.

5.1.2.1 Memory Leak

Leak Canary from Square is a good tool for detecting memory leaks in your app. It creates weak references to activities in your app. (You can also customize it by adding watches to any other objects.) It then checks if the reference is cleared after GC. If not, it dumps heap into a .prof file and analyze it to confirm if there is a leak. If there is one, it shows a notification and in a separate app, it shows the reference tree of how the leak happens. You can find more about Leak Canary in this article: Leak Canary: Detect all memory leaks.