GUI testing for an open source Android project: Wikipedia

Submitted To:
Dr. Xiaoyin Wang
Department of Computer Science

Submitted By:
Deepti Gupta

Contents:

Introduction: In this project, the Espresso Test Recorder tool creates UI tests for Wikipedia application without writing any test code. By recording a test scenario, I recorded interactions with a device and add assertions to verify UI elements in particular snapshots of Wikipedia application. Espresso Test Recorder then takes the saved recording and automatically generates a corresponding UI test. Espresso Test Recorder writes tests based on the Espresso Testing framework, an API in the Android Testing Support Library. The Espresso API encourages creating concise and reliable UI tests based on user actions. By stating expectations, interactions, and assertions without directly accessing the underlying app's activities and views, this structure prevents test flakiness and optimizes test run speed.

GUI Testing: In this project, I tested five major features of this application: Searching for Articles, searching for keywords, jumping to sections, Managing reading list (generating list, add articles in the list, delete the articles from the list), Settings. Figure 1 shows the android emulator, when I created virtual device nexus one.

Figure 1

i. Search for Article: Espresso tests consist of two primary components: UI interactions and assertions on View elements. UI interactions include tap and type actions that we can interact with our application. Assertions verify the existence or contents of visual elements on the screen. For example, an Espresso test for the Notes testing app might include UI interactions for clicking on a button and writing a new note but would use assertions to verify the existence of the button and the contents of the note.

To start recording a test with Espresso Test Recorder, proceed as follows:

1. Click Run > Record Espresso Test.
2. In the Select Deployment Target window, choose the device nexus one. Click OK.
3. Espresso Test Recorder triggers a build of project.

After that, I searched for article and wrote on search window "Wikipedia", which is shown in figure 3 and received the search page.

Add Assertion: For the assertion, I added the result of search article Wikipedia and tested my test case. I tested multiple times and all of time this test was passed. The result is shown in figure 5.

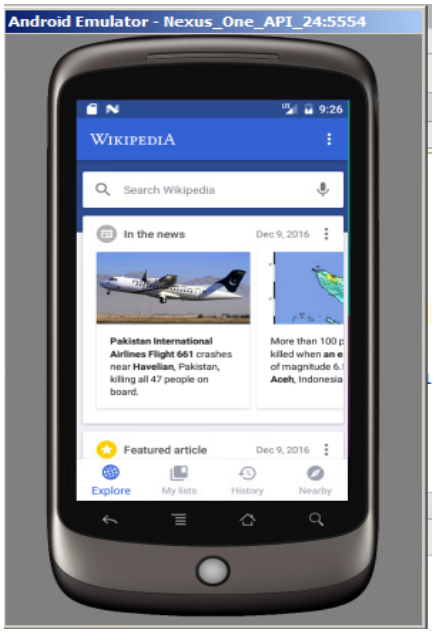

Figure 3

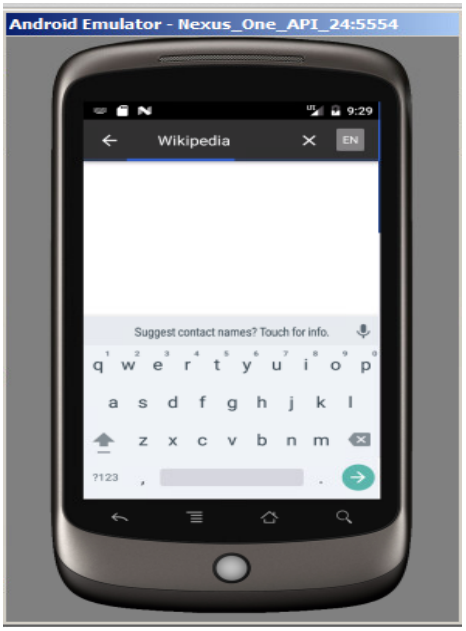

Figure 4

Figure 5

ii. Search for Keywords: I recorded for search for keywords, open an article on Wikipedia explore related to flight information and clicked right for find in page. Here, I typed "flight" keyword. I found this keyword 10 times.

Add Assertion: I found this keyword 10 times, I added assertion 1/10 for this test case. I tested multiple times and all the time this test was successful. This test steps are given below figures 6,7.

Figure 6

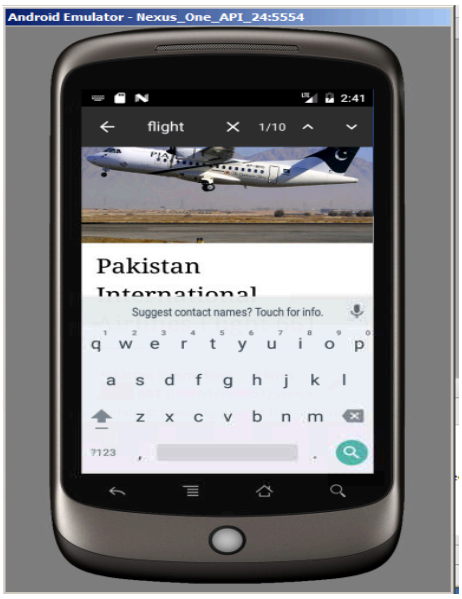

Figure 7

iii. Jumping to sections: For this test, I opened an articles and go through at each sections. I tested this test multiple times and test case was passed. The steps of this test are given in below figures 8,9,10 and 11.

Figure 8

Figure 9

Figure 10

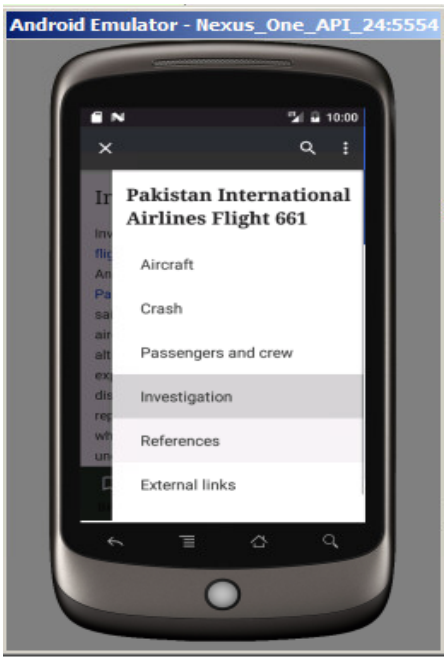

Figure 11

iv. Managing list: In this test case, I generate the reading list and add multiple articles in this list. After adding articles in the reading list and delete the list also.

Add Assertion: For the assertion, I added assertion number of articles, "2 articles" and shown in figure 15. The steps of this test are shown in figure 12,13,14,15,16,17,and 18. This test case run multiple times and passed.

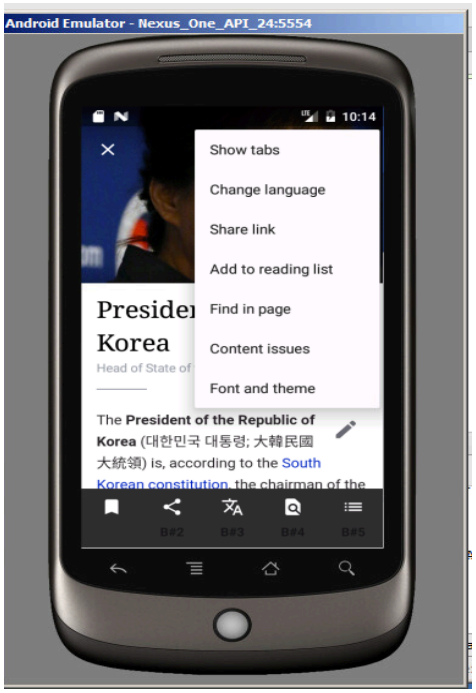

Figure12

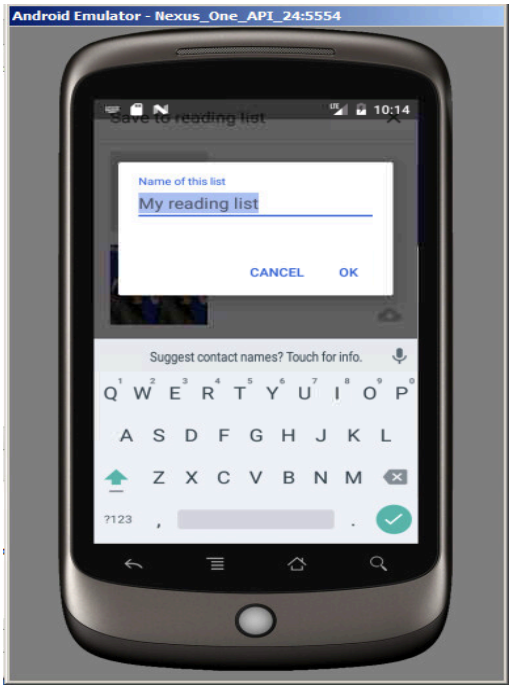

Figure 13

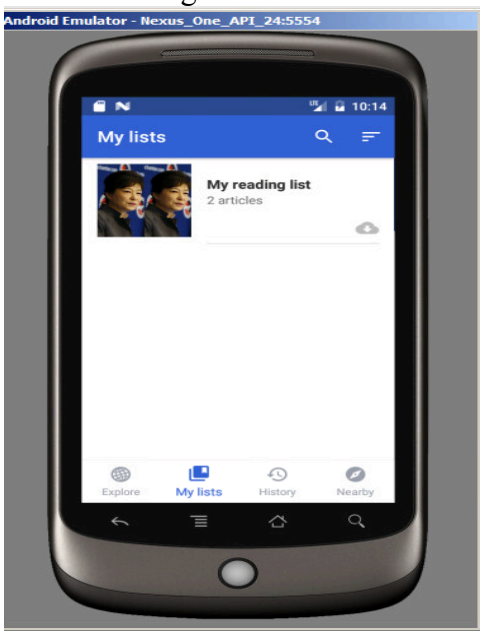

Figure 14

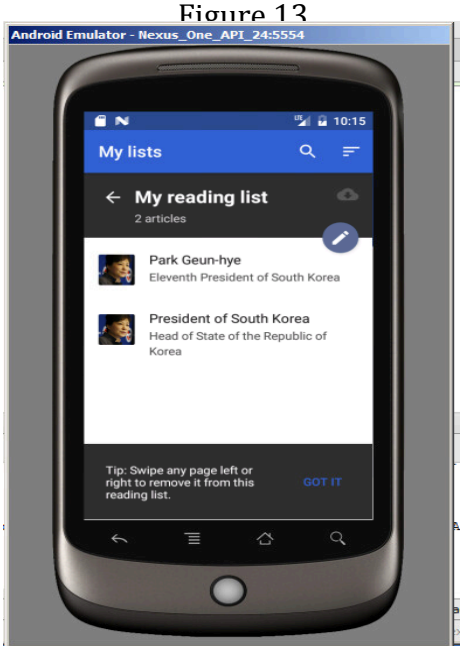

Figure 15

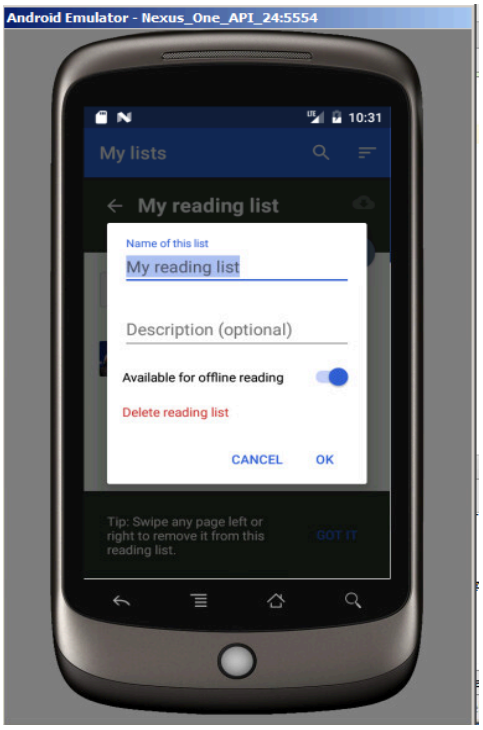

Figure 16

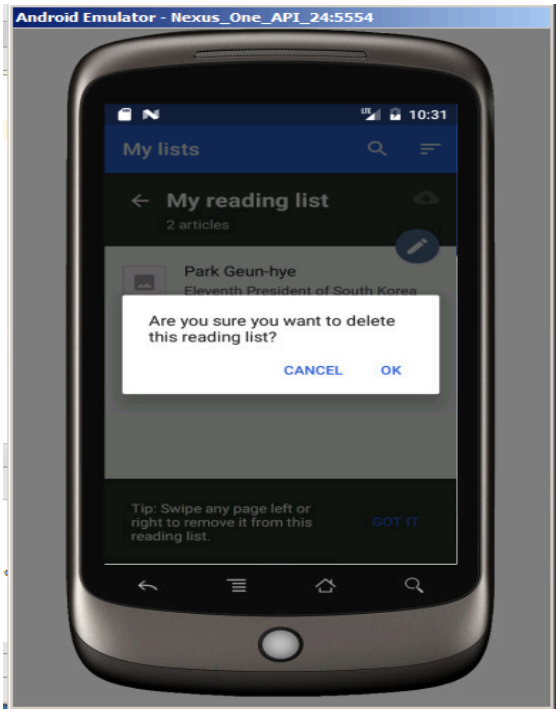

Figure 17

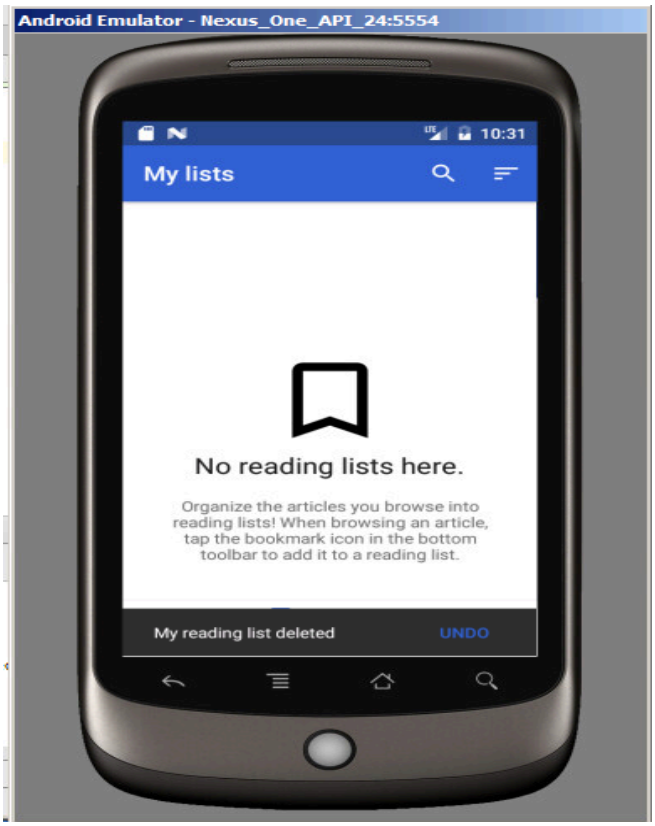

Figure 18

V. Settings: In this test case, I change in the settings. I off the button of show image and then check the articles with no image. After that I clicked on the button of image and check the image of the article.

Add Assertion: For the assertion, I added assertion image of the article and shown in figure 23. The steps of this test are shown in figure 19,20,21,22, and 23. This test case run multiple times and passed.

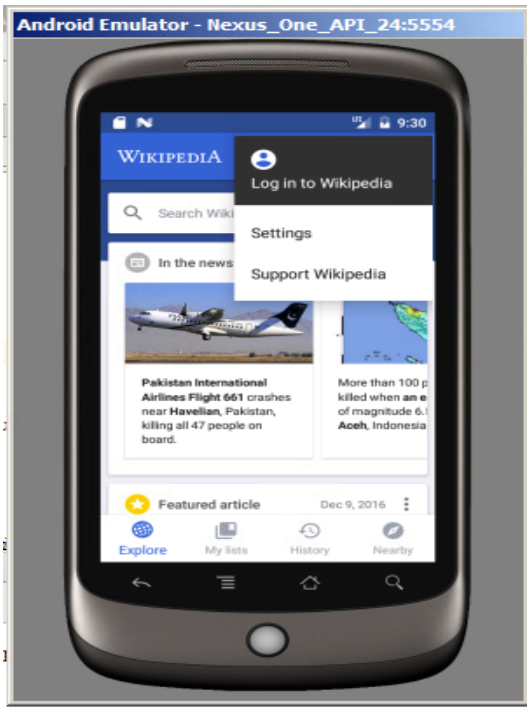

Figure 19

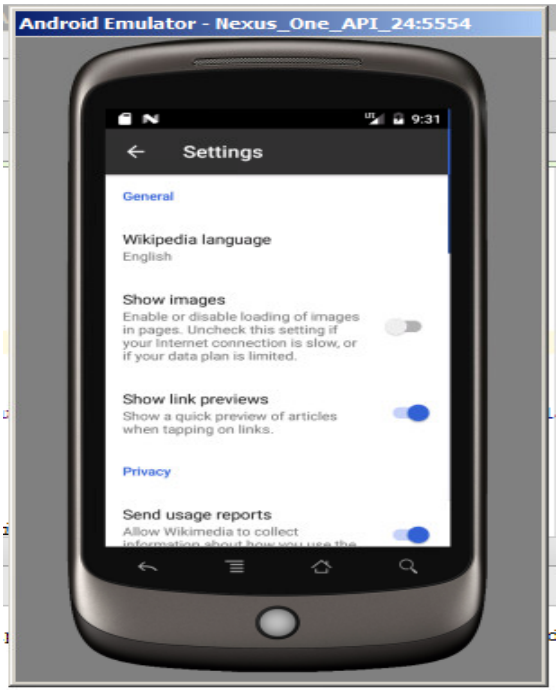

Figure 20

Figure 22

Figure 21

Figure 23

Conclusion: I performed GUI testing and all the test cases run multiple times. I also added assertion with all test cases. All the test cases are passed.

Acknowledgement: I would like to thank Dr. Wang. He helped me a lot to explain this project. I learnt a lot from this project and enjoyed to do testing. In the future, I would like to do more automation testing.

References:

[1] https://developer.android.com/studio/test/espresso-test recorder.html