

Robotium Tests

Introduction

This Android application is very user experience intense in terms of dynamic data. It is not one of the most beautiful applications nor is it the most 3d intense application.

This is no excuse for not doing good graphical tests though. I have created several tests to see if the application does what we want it to do – to follow our requirements. For more comprehensive information about what we require of our application please see the Definition of Done and Release Procedure for example. They are located in the same folder as this document, it should be /doc/.

The tests are based on our User Stories. The main thing I wanted to do with these tests was to make a proof of concept that, firstly, it works. Secondly that testing is required to maintain a good code even though the application isn't that graphically intense. And lastly, that Robotium is a great complement to normal JUnit testing when developing Android applications.

I could have made many more tests but since this is a PoC I don't feel the importance in doing very complex and hard test cases. This is also because the application is very often refactored and that means that I have to change large parts of the tests (due to changes in the xml files that represents the graphical layout)

This document has one heading for each .java file and describes the test that it does and to which User Story it is linked to. All of the test .java files are in the sub-project tests (./tests/) they're all in the same package (se.turbotorsk.mybar.tests). I know that we haven't covered every User Story but since this is the first time I'm using Robotium I'm feeling very satisfied with the result. Some tests are also needed to make sure that the application is performing well, those tests are more linked to our Definition of Done rather than a unique User Story.

Thoughts

It would have been fun to do more complex tests but it took really long time to get Robotium to work at first and therefore there wasn't that much time to do tests. I decided to put more effort in finalizing the project as a whole instead. For example it would have been fun to do a test that adds some spirits, adds favorites, looks if I can make any drinks out of the spirits I have added and then open some kind of intense view. Just to see if the application crashes.

Robotium is quite modular so that I could slice up larger tests into smaller more precise tests, which must be better anyways. I'm just saying that it would have been fun to do some weird tests.

MyBarRobotiumBasicTests

test1StartApplication

This basic test starts the application and makes sure that the right activity is started. This test is not linked to a specific User Story, rather linked to the project as a whole. We can't test that much if the application doesn't start. This is also why this is the first test.

test2ChangeTabToCollection

This test is linked to the first parts of User Story ID 106, Browse the Collection, the parts where we fetch the information. The test makes sure that the Collection activity is loaded correctly and doesn't crash or timeouts. You could say that this tests makes sure that our database backend doesn't crash our frontend.

test3OpenAddSpiritsButton

This test is linked to User Story ID 101 and makes sure that the application can open the add spirits activity in a correct manner. The test presses our option/menu button and presses the button Add Spirits. If everything works as it should the new activity should start. This test checks if that is the case.

test4OpenSettingsButton

This test is linked to User Story ID 107 and is about the same as add spirits button. It opens the settings activity and verifies that it started correctly.

test5OpenAboutBox

This test is linked to User Story ID XXX (unfortunately we don't have any User Story to this test) and is exactly the same as test 3 and 4. Only difference is that it tries to open the about box instead of settings or the add box. There is not so much to say about this test more than it is needed to make sure that the application complies with our demands and expectations.

test6CheckMemory

This test is just a couple of LoC, but it is so important to us. This test checks the memory status on the device. What this means is that if the application would fail this test we would have big problems. We would have had to refactor large parts of the application to make it load the dynamic data more efficient, this would not been easy to fix. Thankfully our application have never failed this test!

test7OpenDrink

This test is linked to User Story ID 103. This is one of the big ones. This test makes sure that we can open drinks. It tests a lot. It tries to open a drink and makes sure that all information is loaded correctly.

You could say that this test isn't done yet because we aren't checking that the text and such is correct. What we do test is to make sure that it doesn't crash. I haven't yet found a way to select the first, or any, drink in the list instead of pressing a static one. But this will have to do for now. I got it to work sometimes with dynamic testing but not well enough. Robotium is created to save time, to get rid of all the manual testing, therefore I don't see this test acceptable yet.

MyBarRobotiumAddFavTest

testAddFavoriteAndVerify

This method is not yet implemented. But it is prepared for.