Testing Summary

The testing phases of the project are based on both manual and code testing. Both two parts consist of black box and white box testing to ensure the project is being delivered under high quality. Here is a brief summary to record methodology, challenges encountered during test stage as well as the responding solution under the situation.

Manual testing

For the black box testing part, most of the behavioral tests are done by developers. In unit and integrated scale, two team members test the others' modular function by operating in android simulator. And both two developers participate in the final system scale testing on android simulators with different operating system version.

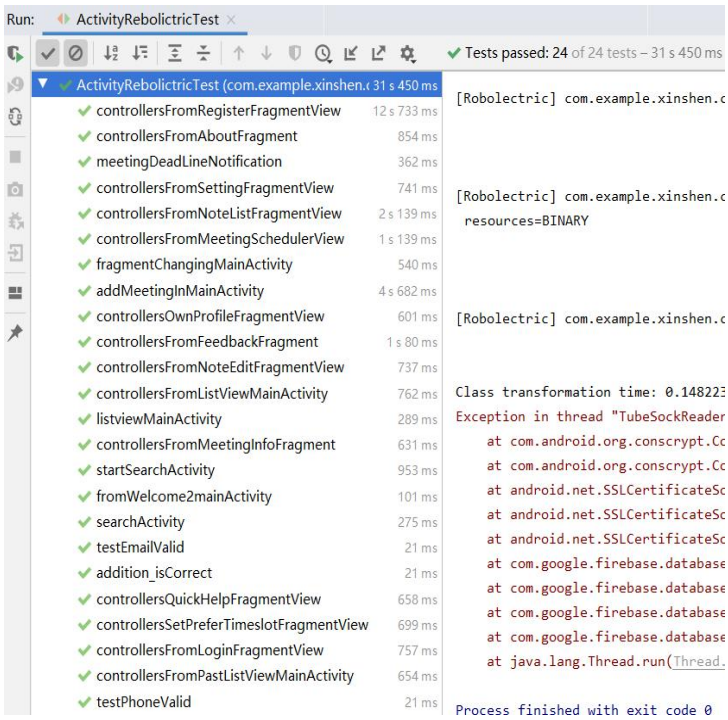
Code testing and selecting libraries

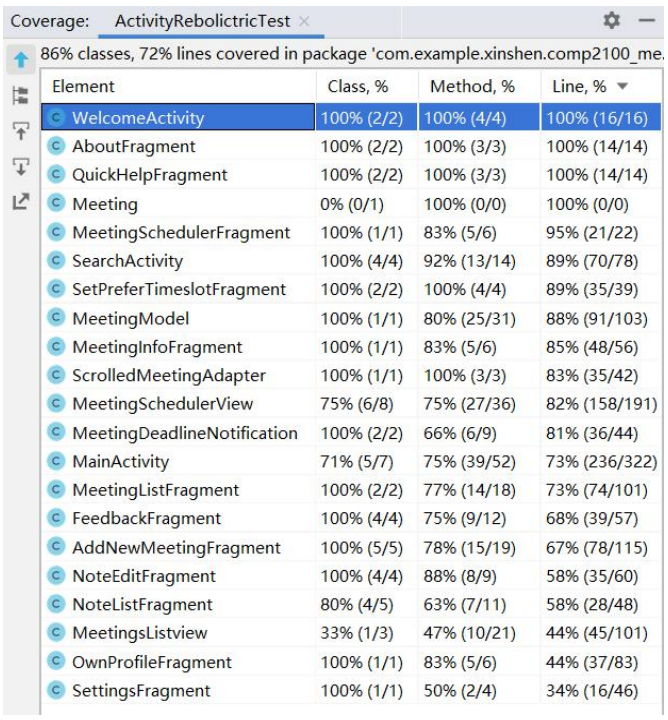
In aspect of white box testing, we mostly apply the test cases at integrated testing level with the help of Rebolictric testing libraries. In the original stage after finishing and passing the unit tests for basic back end logical functions based on Junit4, we have meet the problem that there are a great amount of UI and fragment modules which are hard to be covered by base unit tests. So in order to improve the testing coverage percentage as well as to test the project more thoroughly, we tried three extra popular android testing framework which are Mockito, Espresso and Rebolictric. Having both UI and back end testing supports and being able to generate coverage report automatically, we decide to apply Rebolictric as third party testing supporting libraries. To better calculate the coverage of testing , we gather all testings methods under

Junit testing folder rather than split to instrumentation path.

Testing results and feedback

After developing and running the test cases, we indeed explored many deeply hidden dysfunction such as the conflict with log in and notes function, the unstable life circles for some specific fragments etc. Finally, we manage to pass all expected tests and achieve the testing coverage with 86% classes and around 75% methods covered. Considering the scale of the project, including several fragments, local and server database interaction, MVVM models, the coverage rate and results are satisfied. Some screenshot of test phase as below:





The full testing report with detailed information can be found and accessed within ‘GeneratedTestReport’ folder in the same path of theis testing summary pdf.