

Test Report

The details of test case could be seen in the Test Case File in this directory.

This report will introduce the software testing we did during the whole development stage. We followed test-driven development methodology in this stage to achieve the agile method. "Test-driven development" refers to a style of programming in which three activities are tightly interwoven: coding, testing (in the form of writing unit tests) and design (in the form of refactoring). It can be succinctly described by the following set of rules:

1. write a "single" unit test describing an aspect of the program
2. run the test, which should fail because the program lacks that feature
3. write "just enough" code, the simplest possible, to make the test pass
4. "refactor" the code until it conforms to the simplicity criteria
5. repeat, "accumulating" unit tests over time

Testing in our project could be divided into four stages: unit test, integration test, system test and acceptance test.

Unit Testing

The unit testing was every helpful during the software development. It could both test the new function we added into the system and check if the new function influences existing parts in system. We had unit tests for each module to test every method in new class and new items and widgets in the layout. We firstly wrote the test plan of every part. Then we wrote the code to pass the current test cases. No new code will be added until corresponding tests were made. and no new tests were added until all the existing tests were passed. The test cases were recorded by our members in the test case.

Sub-system integration Testing

The purpose of this kind of testing is to guarantee the correct linkage of multiple modules as a sub-system. We did the sub integration test in the end of every stage in the development because our system is quite complex and it is necessary to make sure that different parts could work together robustly. The details of test cases are in the appendix F. After this part, we found some conflicts of different modules and fixed them successfully. For example, when we merged the data processing module into the main system, some public variables were missing so the program returned an error. The test helped us find out and fix it.

Release Testing

Release testing (also called system integration test) is in the end of product's lifetime. It checks if the whole system could work correctly and product achieves all the goals we expected initially. During system test, we used real scenarios to check if product could work but not just in the theory. We took our wearable devices and mobile phone to the gymnastic and outdoors and did

some exercises to test the capturing ability of our system. We also used our product to record heart rate for a long time as the stress test. The results showed that our product could capture the body data correctly and exported data could be used in final year project.

Acceptance Testing

It is the final test during the development. Acceptance test shows if our product satisfies the requirement of our stakeholders. We demonstrated our project to our stakeholders and got positive reflections and some suggestions. Then we improved our product according to the feedback. We achieved the points that stakeholders want our product to do and added several additional functions. The acceptance test showed that stakeholders were satisfied with the new functions, so we will include them in our final product.