

How your tests are organized? (in a folder per feature, or whatever)

Our tests are organized in terms of the requirements posted in the project outline. Since we are only required to produce test cases required for the front-end behaviour only, we have broken down the requirements into 8 different actions a user can do. Each of these 8 actions will have specific test cases to make sure that the transaction behaves in the intended way with any given input from a user. In order to keep all our test cases organized we will have 8 folders each clearly named with the type of action that the test cases inside are used to test (login, logout createacct, etc.). Each of our test cases will be named in relation to their Test Number. For example, Requirement 10 Test 11 will be abbreviated to R10T11. Let's say test case R4T1 fails when we look into the run and we can easily reference back to our table of test cases and know exactly what functionality of our program that test case is specifically testing.

How you intend that they will be run? (as shell scripts, Windows batch files or whatever)

Since we are developing our project in Java, we are intending for our test cases be run using JUnit. We feel that JUnit is the easiest and most efficient way to run our test cases because JUnit tests can be run automatically, and they check their own results. With using JUnit for our testing, we do not need to manually look through reports of test results to determine if the tests were successful or not, they are simply colour coded and easily identifiable which test cases ran successful (coloured Green) and which test cases did not (coloured Red).

How you intend that the output will be validated? (by comparison, by hand, etc.)

Since we will be using JUnit, the framework itself does all the validation for the output of the methods. In our tests we will include many `assertFalse()` and `assertTrue()` statements to validate what a given output of a method or set of functions should output on a given input. In our preferred IDE of Eclipse, the JUnit plugin will create a new pane when we run our tests showing the test results for each testcase we will create. Once the running of testcases has completed, in the JUnit panel it will display a green bar if all of the test cases passed, or a red bar if any of the test cases have failed validation. This bar will help us quickly determine if we are on the right track in development or not. With the Use of JUnit in development we are able to view the details about the testcases that failed by clicking on the failed test case name and looking in the details pane.

How will results be stored and organized for reporting and comparison with future runs?

Since we are intending on using JUnit to implement and run our Test Cases, we will be able to use the "Test Run History" which allows us to utilize this menu to display a history of previously run tests on our programs. In Eclipse, the JUnit plugin will be cleared every time

eclipse is shut down. To overcome this issue, we can use the XML Import/Export feature that allows us to keep all of our test history in an XML File. We will store these files in a directory and each file will be named using the date and time that the XML file was exported at. For example, if we exported a file on September 27th at 4:30PM the file name would be "09-27-19 16:30.xml" given our created naming scheme. We can use this history menu to compare previously run tests and make educated decisions on if we are spending our time efficiently on particular parts of our project. The "XML Import/Export" feature is also useful because we can import our previous XML files that we have exported and use the "Test Run History Menu" to compare up to 99 (by default) previous Test runs.