Test Plan

We originally decided on the hasRole method from within the User.java file in order to generate the first five test cases of the project. Such reasons for this decision include that hasRole is not a getter or setter method, is not a constructor, is a public method, and calls various other methods that we believed would make the testing more interesting. However, as further discussed later in the report, we discovered afterwards that the User.java file had many various types of dependencies that made development of drivers and testing of the methods extremely difficult. Especially difficult dependencies to get past include those that were required by other java files that contained methods called within User.java. These dependency difficulties were not originally observed when choosing the hasRole method within User.java, as the OpenMRS repository documentation, as well as the User.java file itself, did not document these necessary dependencies. In the end, we found different methods and developed different test cases due to these difficulties. Nevertheless, the original test cases are as follows:

Test Case ID: 1
File Being Tested: ../project/openmrs-core/api/src/main/java/org/openmrs/User.java
Method Being Tested: hasRole
Test Case Driver: ../testCasesExecutables/testCase1Driver.java
Test Data: Doctor false
Expected Result: true

Test Case ID: 2
File Being Tested: ../project/openmrs-core/api/src/main/java/org/openmrs/User.java
Method Being Tested: hasRole
Test Case Driver: ../testCaseExecutables/testCase2Driver.java
Test Data: Doctor true
Expected Result: true

Test Case ID: 3
File Being Tested: ../project/openmrs-core/api/src/main/java/org/openmrs/User.java
Method Being Tested: hasRole
Test Case Driver: ../testCaseExecutables/testCase3Driver.java
Test Data: Doctor false
Expected Result: false

Test Case ID: 4
File Being Tested: ../project/openmrs-core/api/src/main/java/org/openmrs/User.java
Method Being Tested: hasRole
Test Case Driver: ../testCasesExecutables/testCase4Driver.java
Test Data: Nurse true
Expected Result: false

Test Case ID: 5
File Being Tested: ../project/openmrs-core/api/src/main/java/org/openmrs/User.java
Method Being Tested: hasRole

Test Case Driver: ../testCasesExecutables/testCase5Driver.java
Test Data: Nurse true
Expected Result: true

Note that some of the above test cases show differing expected results for the same input. This was done under the assumption that the role of being a superuser could be given and/or taken away before each test run. However, we as a group now realize that this is not a great strategy for software development testing, as a given input should always have the same output for this type of method. This mistake was not repeated within our current test cases and final project results.

As previously mentioned, we later found a different java file, ThreadSafeCircularFifoQueue.java, within the OpenMRS repository code that did not contain such difficult dependencies. This was used for the generation of all five methods and eventual twenty-five test cases. All the finalized selected methods of the project are as follows:

Tested methods:

1. toArray
2. add
3. addAll
4. contains
5. containsAll

Directory of ThreadSafeCircularFifoQueue.java:
~/openmrs-core/api/src/main/java/org/openmrs/util

Testing process
Testing is done on the unit level. Testable methods are selected and tested from a test suite that are stored as individual testing text files as part of a collection of test suites. A bash script, found within the scripts directory, runs all test case text files. Testing is conducted primarily through use of test case drivers that run and compile the code in conjunction with input/output from the bash script file, which is called to run from the command line interface.

Testing schedule
Testing is conducted with an on-demand testing schedule. At the discretion of the tester/developer, a round of testing is initiated. Testing results are output to an html output file that is created, populated, and automatically opened in a browser every time that the script is run. The html output file contains both the time and date of each test run for documentation purposes.

Hardware and software requirements

Such requirements include basic computer and internet access, a Linux virtual machine, a text editor, a basic internal tool suite (command line, etc.), and Java software in order to run the drivers and OpenMRS java files.