Team: 404-Error
Members: Ricky Ramos, Matthew Schwarz, Matthew Kay
October 31, 2017

Deliverable #3

Project: OpenMRS

Task: Design and build an automated testing framework that you will use to implement your test plan per the **Project Specifications** document.

Progress:
Our team first wrote five test cases for the compareNaturalAscii(String s, String t) method within the NaturalStrings class

testCase001.txt

```
# Test cases are written in the following format:
# 1. Test Case ID
# 2. Requirement being tested
# 3. Component being tested
# 4. Class
# 5. Driver being used
# 6. Test Inputs(s)
# 7. Expected Result
#

001
This is testing that the compareNatural method will return a 0 if the two strings are equal
compareNaturalAscii(String s, String t)
NaturalStrings.java
NaturalStringDriver.java
test, test
0
```

testCase002.txt

```
# Test cases are written in the following format:
# 1. Test Case ID
# 2. Requirement being tested
# 3. Component being tested
# 4. Class
# 5. Driver being used
# 6. Test Inputs(s)
# 7. Expected Result
#

002
Testing that the compareNatural method will return a negative number if s lexicographically precedes t
compareNaturalAscii(String s, String t)
NaturalStrings.java
NaturalStringDriver.java
test, white
-3
```

testCase003.txt

```
# Test cases are written in the following format:
# 1. Test Case ID
# 2. Requirement being tested
# 3. Component being tested
# 4. Class
# 5. Driver being used
# 6. Test Inputs(s)
# 7. Expected Result
#

003
Testing that the compareNatural method will return a positive number if s lexicographically follows t
compareNaturalAscii(String s, String t)
NaturalStrings.java
NaturalStringDriver.java
test, string
1
```

testCase004.txt

```
# Test cases are written in the following format:
# 1. Test Case ID
# 2. Requirement being tested
# 3. Component being tested
# 4. Class
# 5. Driver being used
# 6. Test Inputs(s)
# 7. Expected Result
#

004
Testing that the compareNatural method will return a positive number if s lexicographically follows t
compareNaturalAscii(String s, String t)
NaturalStrings.java
NaturalStringDriver.java
test, check
17
```

testCase005.txt

```
# Test cases are written in the following format:
# 1. Test Case ID
# 2. Requirement being tested
# 3. Component being tested
# 4. Class
# 5. Driver being used
# 6. Test Inputs(s)
# 7. Expected Result
#

005
Testing that the compareNatural method will return a positive number if s lexicographically follows t
compareNaturalAscii(String s, String t)
NaturalStrings.java
NaturalStringDriver.java
test, pneumonia
4
```

Next, our group wrote one driver for all the test cases.

NaturalStringDriver.java

```java
package org.openmrs.util;

import java.text.Collator;
import java.util.Comparator;

public class NaturalStringDriver{
  public static void main(String args[]){
    NaturalStrings testNS = new NaturalStrings();
    String[] temp = args[0].split(", ");
    String s = temp[0];
    String t = temp[1];
    System.out.println(Integer.valueOf(testNS.compareNaturalAscii(s, t)));
  }
}

```

After running the runAllTests.sh script, the results are shown in the terminal.

```
matthew@matthew-VirtualBox:~/404-Error/TestAutomation/scripts$ bash runAllTests.sh
/home/matthew/404-Error/TestAutomation/testCases
--------------------
Test: 001
--------------------
Requirement: This is testing that the compareNatural method will return a 0 if the two stri
ngs are equal
Method: compareNatural(String s, String t)
Class: NaturalString.java
Driver: NaturalStringDriver.java
Inputs: test, test
Expected Output: 0
Result: 0

--------------------
Test: 002
--------------------
Requirement: Testing that the compareNatural method will return a negative number if s lexi
cographically precedes t
Method: compareNatural(String s, String t)
Class: NaturalString.java
Driver: NaturalStringDriver.java
Inputs: test, white
Expected Output: -3
```

How-To:

1. Clone the repository
2. Navigate to the scripts folder (TestAutomation/scripts) using the terminal
3. Type “bash runAllTests.sh” in the terminal and hit enter
4. The results of all the tests should be displayed in the terminal
5. The results should also be displayed in your browser

Team Experience:

Overall, this deliverable was a headache but a great learning experience. We can all agree that writing the bash script was the most difficult task. We struggled for hours thinking about how to write it because none of us had that much experience with bash. The issue that we primarily ran into was getting the actual java program to execute once it was compiled. We had issues where the java class was not being located when trying to run. After talking to Professor Bowring about our issue, he informed us about specifying the classpath to run. We found that we were trying to run the program with the classpath being inside of the package, and that was causing issues. We had to have the classpath in the root directory of the package, not the sub-directory containing the program like we previously had. Then when we tried to run the java class using the package “ie org.openmrs.util.NaturalStringDriver” it correctly ran and output the proper result to the bash variable $results. We then used this result to compare to the expected result, and if they were the same then we made the html say that it passed, and if it didn’t it would say it failed. We output to a html report the values in our testcases.txt files, and we then output the result of running the test and whether or not it passes. After all of the test cases run and are input to the html report, we then open the html report in a browser and the results of our tests are displayed in the browser.