Deliverable 3:

Experiences: After the process of planning, it was now time to work on some actual code. Our group had no issues generating outputs from the methods we chose to test but we did have issues capturing the outputs and displaying them in an easy to read fashion. First, we tried to pipe the results from the chosen methods to a text file but for whatever reason we couldn't get it to work. Instead, we wrote a program that writes the results of each method to a text file with other details from each test. Then, the relevant data from the text file is written and formatted into the results HTML file in a way that is easy to read and understand. The work we had to learn the most for was working with the java packages system. We had not had experience with this system before and needed to use it to have a properly organized framework rather than throwing all java files into one directory.

Test Case Execution: For running the testing framework, we have a python script for iterating through each test case. Java and python are required for running the tests; there have been no issues with newer versions. The python script is executed by typing "`python scripts/runAllTests.py`" into the command line inside the TestAutomation directory of the project. If the java files have not been compiled then the string "`c`" can be added as an argument to compile the java files necessary to run the code. The script will retrieve the command to launch the driver from each test case file, the drivers launched will record the results of the test in the text file. New test cases are easy to implement as long as they follow the proper test case

template we have designed; the script requires no other information. The script then grabs the new data from the text file and writes it to the results HTML file, where it formats it in a way that is easy to read and understand.

Architectural Description: The test framework script is executed from the TestAutomation folder, which is the top directory. The controlling python script is in the scripts directory, and it has to navigate to the testcases directory and to each line of each file in that folder to find the commands to execute each method being tested with its specific parameters. The testcasesexecutables directory contains auxiliary code that executes the tests on the methods and writes the results back to the files. The script then collects this data and writes to a html file in the reports folder which is automatically opened by the script when all tests are complete. The docs file has the README file in it with the instructions on running the testing code. The project directory contains the required classes from the original project for the methods being tested.

Example Output: The following is an example of how the results of the testing framework will be output, using the first five tests. The chart displays all relevant information from the files in a labeled chart. Note that direct file paths will differ on each machine.

Tanaguru TestAutomation Results

Test ID	Tested class	Tested Method and Parameters	Command for executable	Expected outcomes	Last Result from Running Test	Success or Fail
calculate_1	DistanceCalculator.java	calculate(Color(255,255,255), Color(0,0,0))	java -cp . testcasesexecutables.TestCalculate 255 255 255 0 0 0 441.67 /home/osboxes/Team-6/TestAutomation/testCases/calculate_1.txt	441.67	367.77	Fail
calculate_2	DistanceCalcualtor.java	calculate(Color(0,0,0), Color(255,255,255))	java -cp . testcasesexecutables.TestCalculate 0 0 0 255 255 255 441.67 /home/osboxes/Team-6/TestAutomation/testCases/calculate_2.txt	441.67	367.77	Fail
calculate_3	DistanceCalculator.java	calculate(Color(0,0,0), Color(0,0,0))	java -cp . testcasesexecutables.TestCalculate 0 0 0 0 0 0 0.0 /home/osboxes/Team-6/TestAutomation /testCases/calculate_3.txt	0.0	0.0	Pass
calculate_4	DistanceCalcualtor.java	calculate(Color(255,200,34),Color(255,255,17))	java -cp . testcasesexecutables.TestCalculate 255 200 34 255 255 17 57.57 /home/osboxes/Team-6/TestAutomation/testCases/calculate_4.txt	57.57	54.45	Fail
calculate_5	DistanceCalculator.java	calculate(Color(200,7,16),Color(57,57,57))	java -cp . testcasesexecutables.TestCalculate 200 7 16 57 57 57 156.94 /home/osboxes/Team-6/TestAutomation/testCases/calculate_5.txt	156.94	139.77	Fail

For a more in-depth look at the code of the script and drivers, it is available with comments in the TestAutomation directory of our GitHub repository.