Deliverable 3:

Experiences: This section of the project proved to be more difficult because coming up with a plan to create a testing framework and actually building the tests are different processes. 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 driving through each test case. 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. Once the python script has executed the command from the test case files, it updates the results of the test in the text file. The script then grabs the data from the text file and writes it to the results HTML file, while also checking if the file has changed, 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 driving 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 data to other files. The data collected from all of these files being run is collected and written to a file in the reports folder which is automatically opened by the script when it is finished being written to. 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.