Building of the script was pretty straightforward. The script that was written to list the current script's directory previously, served as a good basis for our runAllTests script. Some slight modifications needed to be made to the script, which include compiling and executing the Java file/Driver. The goal of the runAllTests script was to compile and run the Driver class and output the result into an html file.

The Driver itself handles most of the work and is the bulk of the framework. The script will send the correct directory to the driver (the /testCase/ directory for our purposes), and will put this into the args[0] variable. From there, the driver will build an array to store all of the files that need to be analyzed. From there, the framework will split the text document up into separate tokens, and will compare the result to the expected output in order to determine if the test passed or failed.

Update 10/31/17: It was determined that a lack of proper understanding of the instructions led to the incorrect structuring of our testing framework. This will be rectified and detailed in future deliverables.

HOW TO RUN:

1. Installing Java
    - Check if the Java JDK is installed: `java -version`
    - If not, Update libraries with `sudo apt-get update`
    - After libraries are updated, run `sudo apt-get install -y default-jdk`
2. Cloning the TBD Repository
    - Create the folder to put the repository `touch <your_foldername_here>`
    - Navigate to the folder you created with `cd <your_foldername_here>`
    - Initialize the git folder `git init`
    - Clone the repository to your system: run `git clone https://github.com/CSCI-362-01-2017/TBD`
3. Running the Script
    - Navigate to the script's directory: `cd /TBD/scripts`
    - Run the command `./runAllTests`

If done correctly, the script should execute and open a html page with the test cases displayed on the screen.

Tested Items:

Test Case Specification Template

- Test Number or ID - Number of the specific test case to be tested.
- Requirement being Tested - The condition being tested
- Component being Tested - A class within the application will be chosen for a method it possesses for testing.
- Method being Tested - A procedure within the class will be selected for the development and implementation of the test cases.
- Test Input(s) Including Command Line Arguments - This will include the parameters necessary to test the specified requirement, and the command to run this test from the command line.
- Expected Outcome(s) - This will specify the conditions necessary to determine whether the test has passed, after being run.

By the end of the testing period, we will have developed a total of twenty-five test cases. To date, we have the following five test specifications:

1.
    - **Test Number or ID** - 6
    - **Requirement Being Tested** - Tests that if the hour is greater than 23, it should return 8 to represent night.
    - **Component Being Tested** - `org.glucosio.android.tools.ReadingTools`
    - **Method Being Tested** - `public int hourToSpinnerType(int hour)`
    - **Test Input(s) Including Command Line Arguments** - `hour = 24`
    - **Expected Outcome(s)** - `8`
2.
    - **Test Number or ID** - 7
    - **Requirement Being Tested** - Tests that if the hour is greater than 20 but less then 23, it should return 5 to represent after dinner.
    - **Component Being Tested** - `org.glucosio.android.tools.ReadingTools`
    - **Method Being Tested** - `public int hourToSpinnerType(int hour)`
    - **Test Input(s) Including Command Line Arguments** - `hour = 21`
    - **Expected Outcome(s)** - `5`
3.
    - **Test Number or ID** - 8
    - **Requirement Being Tested** - Tests that if the hour is greater than 17 but less then 20, it should return 4 to represent before dinner.
    - **Component Being Tested** - `org.glucosio.android.tools.ReadingTools`
    - **Method Being Tested** - `public int hourToSpinnerType(int hour)`

- **Test Input(s) Including Command Line Arguments** - `hour = 19`
- **Expected Outcome(s)** - `4`

4.
   - **Test Number or ID** - 9
   - **Requirement Being Tested** - Tests that if the hour is greater than 13 but less then 17, it should return 3 to represent after lunch.
   - **Component Being Tested** - `org.glucosio.android.tools.ReadingTools`
   - **Method Being Tested** - `public int hourToSpinnerType(int hour)`
   - **Test Input(s) Including Command Line Arguments** - `hour = 15`
   - **Expected Outcome(s)** - `3`
5.
   - **Test Number or ID** - 10
   - **Requirement Being Tested** - Tests that if the hour is greater than 4 but less then 7, it should return 0 to represent before breakfast.
   - **Component Being Tested** - `org.glucosio.android.tools.ReadingTools`
   - **Method Being Tested** - `public int hourToSpinnerType(int hour)`
   - **Test Input(s) Including Command Line Arguments** - `hour = 5`
   - **Expected Outcome(s)** - `0`