

Introduction:

This document tells how the program will be executed, tested and extended.

Acknowledgement:

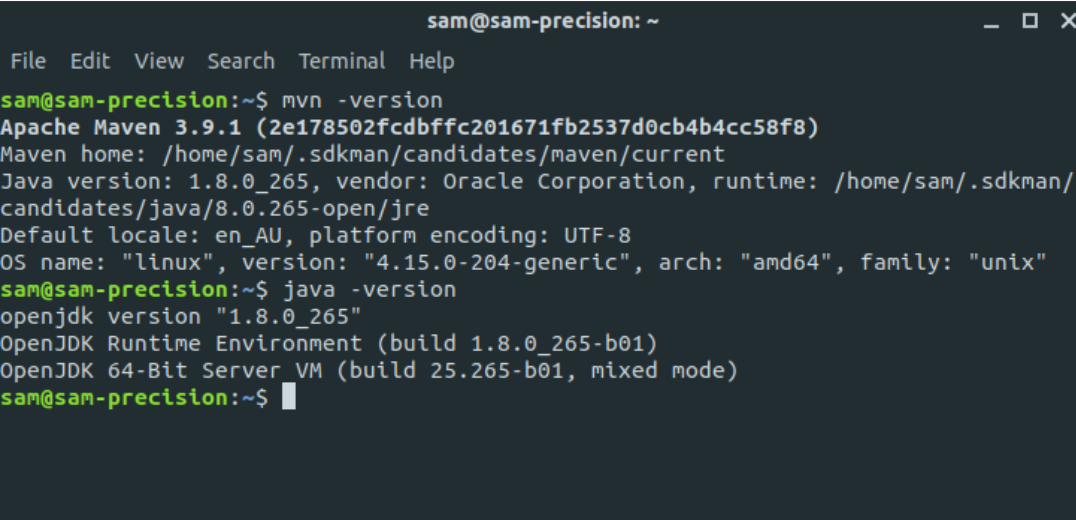
There is room for improvement but because of the time constraints, all the improvements haven't been applied, which have been discussed at the bottom of the report.

Requirements to Run and Test this Project:

- 1-JDK8
 - 2-JDK path is added in the classpath
 - 3-Maven
 - 4-Maven path is added in the classpath
-

Run Using Maven

- 1- Please make sure maven is installed and maven_home is set.
- 2-Please check maven is added in the classpath.
- 3-Open terminal type mvn you should see the version number like below.

A terminal window titled 'sam@sam-precision: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the output of 'mvn -version' and 'java -version' commands. The Maven output shows version 3.9.1 and the Java output shows OpenJDK 1.8.0_265.

```
sam@sam-precision: ~  
File Edit View Search Terminal Help  
sam@sam-precision:~$ mvn -version  
Apache Maven 3.9.1 (2e178502fcd9bfc201671fb2537d0cb4b4cc58f8)  
Maven home: /home/sam/.sdkman/candidates/maven/current  
Java version: 1.8.0_265, vendor: Oracle Corporation, runtime: /home/sam/.sdkman/  
candidates/java/8.0.265-open/jre  
Default locale: en_AU, platform encoding: UTF-8  
OS name: "linux", version: "4.15.0-204-generic", arch: "amd64", family: "unix"  
sam@sam-precision:~$ java -version  
openjdk version "1.8.0_265"  
OpenJDK Runtime Environment (build 1.8.0_265-b01)  
OpenJDK 64-Bit Server VM (build 25.265-b01, mixed mode)  
sam@sam-precision:~$
```

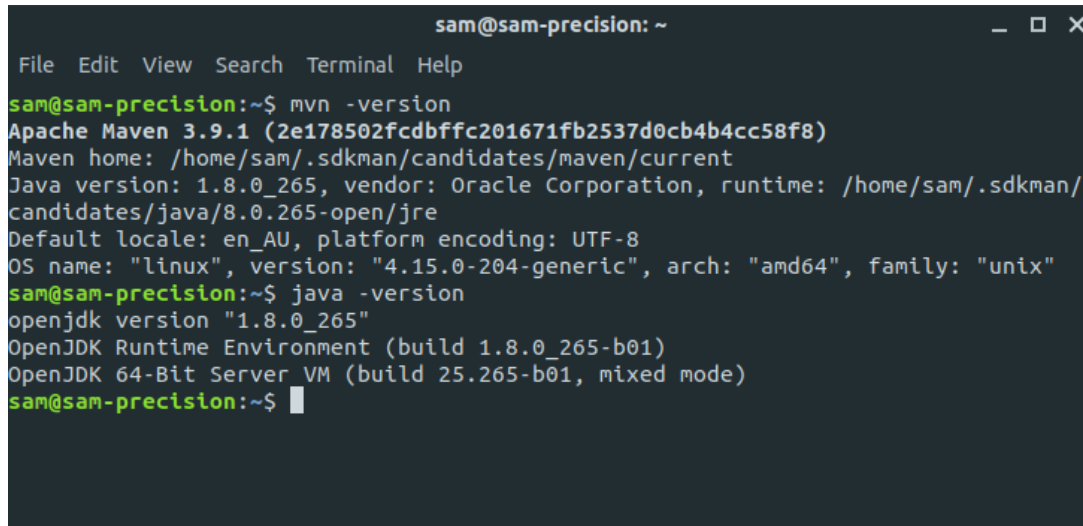
- 4-Type "cd" to the project folder root where there is a pom.xml, is residing

- 5-Run the follow command

```
mvn exec:java -Dexec.mainClass=com.geekheights.app.Main  
-Dexec.args="src/test/java/resources/correctData/sample1.txt"
```

Run Tests Using Maven

- 1-Please make sure maven is installed and maven_home is set.
- 2-Please check maven is added in the classpath.
- 3-Open terminal type mvn you should see the version number like below.

A terminal window titled 'sam@sam-precision: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The user enters 'mvn -version' and 'java -version'. The output shows Apache Maven 3.9.1 and OpenJDK 1.8.0_265.

```
sam@sam-precision: ~  
File Edit View Search Terminal Help  
sam@sam-precision:~$ mvn -version  
Apache Maven 3.9.1 (2e178502fcd8bffc201671fb2537d0cb4b4cc58f8)  
Maven home: /home/sam/.sdkman/candidates/maven/current  
Java version: 1.8.0_265, vendor: Oracle Corporation, runtime: /home/sam/.sdkman/  
candidates/java/8.0.265-open/jre  
Default locale: en_AU, platform encoding: UTF-8  
OS name: "linux", version: "4.15.0-204-generic", arch: "amd64", family: "unix"  
sam@sam-precision:~$ java -version  
openjdk version "1.8.0_265"  
OpenJDK Runtime Environment (build 1.8.0_265-b01)  
OpenJDK 64-Bit Server VM (build 25.265-b01, mixed mode)  
sam@sam-precision:~$
```

- 4-Type "cd" to the project folder root where there is a pom.xml, is residing
- 5-Run the follow command "mvn test -X "
- 6-The output should be similar to below

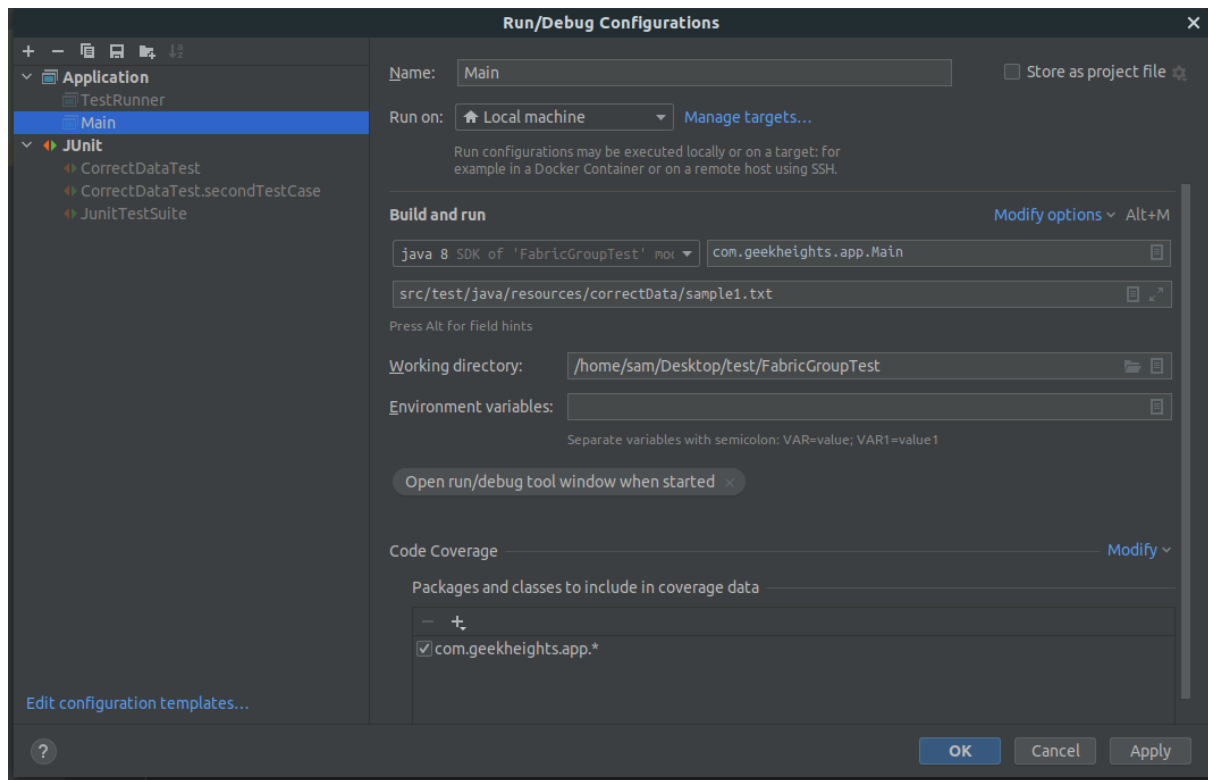
A terminal window showing the output of 'mvn test -X'. It displays test results for 'com.test.fabricgroup.TankerRulesTest' and 'com.test.fabricgroup.CorrectDataTest', all passing.

```
Terminal: Local x Local (2) x + v  
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.025 s - in com.test.fabricgroup.TankerRulesTest  
[INFO] Running com.test.fabricgroup.CorrectDataTest  
Instantiating array list in ParseAndLoadData ...  
2400 5215  
3000 5750  
900 1200  
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.008 s - in com.test.fabricgroup.CorrectDataTest  
[INFO]  
[INFO] Results:  
[INFO]  
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0  
[INFO]
```

Run Using IntelliJ IDE

The program has been built with Maven so it can be imported into any IDE, preferably IntelliJ.

Once the program is opened in an IDE the com.papercut.test.run can be run directly by passing the command line args from IDE like the below image.



Run Tests Using IntelliJ

Tests can be run in a similar way by running “JUnitTestSuite” or “TestRunner” file.

Improvements

There are many improvements that could be made but haven't been done because of the time constraints.

- 1) There is room for better error handling.
 - 2) JUnit 5 could be used.
 - 3) A newer version of java could have been used.
 - 4) Spring batch could have been used (Since haven't worked on it for over a year so needed a bit of a revision).
 - 5) Strategies could be defined in an enum instead of a functional interface.
 - 6) Detailed comments and java docs are missing because of the time constraint.
 - 7) The architecture of the application could have been better especially in the slab area.
 - 8) If I had more time I could have used Drools which is mainly for such situations where you have a set of rules that needs to be applied to data (Since haven't worked on it for over a year so needed a bit of a revision).
-