Jonathan Licht

Aaron Maus

CMPS 1600

October 20, 2020

User Manual

**Running Program**

1. Open the Terminal application.
2. Locate the project in the directory.
   1. Enter “ls” to find where you are in the directory.
   2. Enter “cd” and the name of folder you would like to move into.
   3. Use these two commands to locate the project.
3. Once you are in the correct place in the directory, enter “javac PerfectCandidate.java” into the terminal.
   1. This command compiles the main program so the program can run.
4. Then enter “java PerfectCandidate” into the terminal.
   1. This command actually runs the program.
5. This should open a graphic user interface (GUI).
6. From here, the GUI will prompt you to chose a file.
7. Navigate through your computer to find an input file.
   1. An example of a valid input file is shown below
8. Once you have found and selected an input file, a menu that allows you to cast votes should pop up.
   1. To cast a vote, click on the candidate you would like to vote for, click “Cast Vote”, and then click “Ok”.
   2. Vote for whoever you would like to as many times as you would like.
9. To exit the voting menu, close the window and click “Yes”.
10. The program should prompt you to create an output file.
    1. It is usually easiest to call your output file “output.txt” however you can name your file whatever you would like.
11. At this point the program is done running. To find the results of the ballot, find and open your output file. The results should be presented in a format that is easy to read.

A simpler way of running the program is through the Jar file. In your computer, find the file named “PerfectCandidate.jar” If you double click the file, you do steps one through four. Continue running the program with step five.

**Running JUnit Test**

Running the JUnit tests for the two implemented classes (Candidate and Ballot) requires a process of steps.

1. Follow steps 1 and 2 as listed above.
2. Once you are in the correct place in the directory, test the Candidate class by entering:

javac --class-path .:junit-platform-console-standalone-1.7.0.jar CandidateTests.java

* 1. This command compiles CandidateTests.java so the program can run.

1. Then enter:

java -jar junit-platform-console-standalone-1.7.0.jar --class-path . --

select-class CandidateTests

* 1. This command runs the CandidateTests program.
  2. The output should show whether the tests were passed or failed for the Candidate class.

1. Test the Ballot Class by entering:

javac --class-path .:junit-platform-console-standalone-1.7.0.jar Ballot

Tests.java

* 1. This command compiles BallotTests.java so the program can run.

1. Then enter:

java -jar junit-platform-console-standalone-1.7.0.jar --class-path . --

select-class BallotTests

* 1. This command runs the BallotTests program.
  2. The output should show whether the tests were passed or failed for the Ballot class.

**Example Input File**

An input file should look like this:

Coolest Person Award

4

Jonathan;Rhode Island

Forrest;California

Luke;Oregon

Lukas;Minnesota

The first line is the name of the ballot. The second line is the number of candidates. The next lines are the candidates and their party separated by a semicolon.