|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test1:  PBI #4: Main function of MyMenuDemo accepts command line arguments. |
| Member(s) Responsible | Shi Chen |
| Inputs | 1. Auditrand2.csv is the filename entered on command line. It is located in /testing. 2. SmallElection8.csv is the filename entered on command line. It is located in /testing. |
| Tests | 1. Whether MyMenuDemo.main reads an IR file from command line 2. Whether MyMenuDemo.main reads an OPL file from command line |
| Outputs | Screen summary of file. |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the MyMenuDemoTest.java (see the Readme for instructions on building tests in Intellij). 2. Move SmallElection8.csv and auditrand2.csv from /testing to the root of project. 3. Run MyMenuDemoTest.java |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test2:  PBI #3: IRVoting displays a table of runoff results on the terminal. |
| Member(s) Responsible | Hong |
| Inputs | irTest1.csv is the filename entered on command line. It is located in /testing. |
| Tests | Tests whether the table shows correct results for a standard election file. The function getAllRecords() |
| Outputs | Runoff table is displayed on terminal. |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the IRVotingTest.java (see the Readme for instructions on building tests in Intellij). 2. Move irTest1.csv from /testing to the root of project. 3. RunIRVotingTest.java |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 3:  PBI #4: Main function of MyMenuDemo accepts command line arguments. |
| Member(s) Responsible | Shi Chen |
| Inputs | 1. SmallElection8.csv is the filename entered on command line. It is located in /testing. |
| Tests | 1. Whether MyMenuDemo.main ignores spaces. |
| Outputs | Screen summary of file. |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. Move SmallElection8.csv from /testing to the root of project. 3. Run MyMenuDemoTest.java with argument “ SmallElection8.csv” |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 4:  PBI #4: Main function of MyMenuDemo can alternate between GUI window and command line. |
| Member(s) Responsible | Shi Chen |
| Inputs | None |
| Tests | 1. Whether MyMenuDemo.main ignores command line if no arguments are supplied. |
| Outputs | None |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the MyMenuDemoTest.java (see the Readme for instructions on building tests in Intellij). 2. Run MyMenuDemoTest.java 3. see TestMainNoInput() |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 5:  PBI #4: Main function of MyMenuDemo can alternate between GUI window and command line. |
| Member(s) Responsible | Shi Chen |
| Inputs | None |
| Tests | 1. Whether MyMenuDemo.main ignores command line if no arguments are supplied and the GUI pops up. |
| Outputs | None |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. Move SmallElection8.csv from /testing to the root of project. 3. Run java MyMenuDemo 4. See if GUI pops up. |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 6:  PBI #4: Main function of MyMenuDemo accepts command line arguments. |
| Member(s) Responsible | Shi Chen |
| Inputs | None |
| Tests | 1. Whether MyMenuDemo.main can handle bad input |
| Outputs | None |
| Pass or Fail | Fail |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. Run java MyMenuDemo woops SmallElection.csv 3. see if it fails. |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 7:  PBI #6: A GUI is displayed to the user. Testing the GUI window code. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | None |
| Tests | 1. Tests whether the init() function generates the window. |
| Outputs | None |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the MyMenuDemoTest.java (see the Readme for instructions on building tests in Intellij). 2. Run MyMenuDemoTest.java 3. see TestInit() |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 8:  PBI #6: A GUI is displayed to the user. Testing private myEvent() method which generates the file explorer window. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | None |
| Tests | 1. Tests whether the file explorer window is generated when prompted. |
| Outputs | None |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the MyMenuDemoTest.java (see the Readme for instructions on building tests in Intellij). 2. Run MyMenuDemoTest.java 3. see TestEvent() checks whether the file explorer was activated. |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 9:  PBI #6: A GUI is displayed to the user. Manual testing to see if the GUI is displayed. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | None |
| Tests | 1. Test whether the GUI pops up. |
| Outputs | None |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. Run java MyMenuDemo 3. Click File at the top of the menu. 4. See if the two options are offered. |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 10:  PBI #6: The user can input file name from GUI. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | None |
| Tests | 1. Whether the SearchFile.getPath() is correct. |
| Outputs | None |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile SearchFileTest.java (see the Readme for instructions on building tests in Intellij). 2. Run SearchFileTest.java |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 11:  PBI #6: The user can input file name from GUI. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | None |
| Tests | 1. Whether the JFrame for prompting the user for filename is generated correctly. |
| Outputs | None |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. Run java SearchFile 3. Inspect the Frame. |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 12:  PBI #6: The user can input file name from GUI. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | testing/ irTest1.csv |
| Tests | 1. The user can input files in the root directory. |
| Outputs | Audit file Election\_Wed Dec 05 22:29:00 CST 2018\_Summary.txt is generated |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. Run java SearchFile 3. Inspect the Frame. 4. move irTest1.csv from testing to root directory 5. enter irTest1.csv in text box 6. Click Confirm |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 13:  PBI #6: The user can input file name from GUI. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | testing/SmallElection11.csv |
| Tests | 1. The user can input files in a sub-directory. |
| Outputs | Election\_Wed Dec 05 22:37:05 CST 2018\_audit.txt |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. Run java SearchFile 3. Inspect the Frame. 4. enter testing/SmallElection11.csv in text box 5. Click Confirm. |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 14:  PBI #6: The user can input file name from GUI. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | nonexist.csv |
| Tests | 1. The user can input files in a sub-directory. |
| Outputs | Should be nothing |
| Pass or Fail | Fail |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. Run java SearchFile 3. Inspect the Frame. 4. enter nonsexist.csv in text box 5. Click confirm. |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 15:  PBI #6: The user can input file name from GUI. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | testing/SmallElection11.csv |
| Tests | 1. User can search for file in a sub-directory. |
| Outputs | Election\_Wed Dec 05 22:37:05 CST 2018\_audit.txt |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. Run java SearchFile 3. Inspect the Frame. 4. enter testing/SmallElection11.csv in text box 5. click Confirm |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 15:  PBI #6: The user can look for file on drive. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | testing/SmallElection1.csv |
| Tests | 1. The user can search for a file in a subdirectory. |
| Outputs | Election\_Wed Dec 05 23:05:04 CST 2018\_audit.txt |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. Run java MyMenuDemo 3. Inspect the Frame. 4. Click File at the top 5. Click open from file manager 6. Under testing folder select SmallElection1.csv 7. Open |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 17:  PBI #6: The user can look for file on drive. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | testing/irTest1.csv |
| Tests | 1. The user can search for a file not in project file hierarchy. |
| Outputs | Election\_Wed Dec 05 23:15:08 CST 2018\_audit.txt |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. move irTest1.csv to Desktop 3. Run java MyMenuDemo 4. Inspect the Frame. 5. Click File at the top 6. Click open from file manager 7. Under Desktop select SmallElection1.csv 8. Open |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number | Test 18:  PBI #6: The user can look for file on drive. |
| Member(s) Responsible | Shi Chen, Dong Liang |
| Inputs | buglist.docx |
| Tests | 1. The user can search for a file not in project file hierarchy. |
| Outputs | Nothing |
| Pass or Fail | Pass |
| Date | 11/27/18 |
| Steps | 1. Compile the class files (see Readme for instructions on how to compile system) 2. move irTest1.csv to Desktop 3. Run java MyMenuDemo 4. Inspect the Frame. 5. Click File at the top 6. Click open from file manager 7. Under Desktop select SmallElection1.csv 8. Open |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | Test 19  3, As an election official, I would like to see a table of the election showing each round of the IRV and the number of votes that the candidate added/subtracted for that round. I would like the table should be displayed to the screen so that I can inspect the correctness of the runoffs and understand it easily. |
| Team Member(s) Responsible: | Qing Hong |
| Inputs: | testing/irTest1.csv |
| Tests: | To make sure the information of every step when running IRVoting is written to log file. |
| Outputs: | All Tests Passed |
| Passed or Failed: | Passed |
| Date: | Dec 8th, 2018 |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | Test 20  2, As an election official, I need all ballots in an IR election that do not have at least half the candidates ranked to be invalidated and written to file named invalidated\_dateofelection\_xxx in order to audit the election and because ballots need to have at least half the candidates be ranked to be valid and impact the election. |
| Team Member(s) Responsible: | Song Liu |
| Inputs: | testing/irTest1.csv |
| Tests: | To make sure the info of every invalid ballot when running IRVoting is written to log file and is not counted |
| Outputs: | All Tests Passed |
| Passed or Failed: | Passed |
| Date: | Dec 9th, 2018 |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | Test 21  2, As an election official, I need all ballots in an IR election that do not have at least half the candidates ranked to be invalidated and written to file named invalidated\_dateofelection\_xxx in order to audit the election and because ballots need to have at least half the candidates be ranked to be valid and impact the election. |
| Team Member(s) Responsible: | Shi Chen |
| Inputs: | testing/irTest5.csv |
| Tests: | To test no memory error occurs if all ballots were invalidated. |
| Outputs: | invalidated\_Sun Dec 09 22:01:17 CST 2018.txt, Election\_Sun Dec 09 22:01:17 CST 2018\_audit.txt |
| Passed or Failed: | Passed |
| Date: | Dec 9th, 2018 |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | Test 22  2, As an election official, I need all ballots in an IR election that do not have at least half the candidates ranked to be invalidated and written to file named invalidated\_dateofelection\_xxx in order to audit the election and because ballots need to have at least half the candidates be ranked to be valid and impact the election. |
| Team Member(s) Responsible: | Shi Chen |
| Inputs: | testing/SmallElection14.csv |
| Tests: | Whether removing ballots stops a correct runoff from happening. |
| Outputs: | invalidated\_Thu Dec 06 23:27:29 CST 2018.txt  Election\_Thu Dec 06 23:27:29 CST 2018\_audit.txt |
| Passed or Failed: | Passed |
| Date: | Dec 6th, 2018 |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | Test 23  2, As an election official, I need all ballots in an IR election that do not have at least half the candidates ranked to be invalidated and written to file named invalidated\_dateofelection\_xxx in order to audit the election and because ballots need to have at least half the candidates be ranked to be valid and impact the election. |
| Team Member(s) Responsible: | Shi Chen |
| Inputs: | testing/SmallElection10.csv |
| Tests: | Whether removing ballots slow down an election for a large ballot file. |
| Outputs: | invalidated\_Thu Dec 06 22:47:40 CST 2018.txt,  Election\_Thu Dec 06 22:47:40 CST 2018\_audit.txt |
| Passed or Failed: | Passed |
| Date: | Dec 6th, 2018 |

|  |  |
| --- | --- |
| The PBI, the Task Description (from Sprint Log) with Unique Testing Number: | Test 24  2, As an election official, I need all ballots in an IR election that do not have at least half the candidates ranked to be invalidated and written to file named invalidated\_dateofelection\_xxx in order to audit the election and because ballots need to have at least half the candidates be ranked to be valid and impact the election. |
| Team Member(s) Responsible: | Shi Chen |
| Inputs: | testing/irTest6.csv |
| Tests: | Whether removing ballots stops breaking of ties. |
| Outputs: | Testing/invalidated\_Sun Dec 09 22:16:17 CST 2018.txt  Testing/Election\_Sun Dec 09 22:16:17 CST 2018\_audit.txt |
| Passed or Failed: | Passed |
| Date: | Dec 9th, 2018 |

Below Are All the Old Tests, which we re-ran regressively. Because there are a lot of them, the tester (mostly Shi) did not write a new log for them. They’ve been edited in the old logs.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi01** | **Name(s) of Testers: Shi Chen** |
| **Test Description: System testing for inputting an OPL file with 2 ballots.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection1.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection1.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection1.csv from testing directory into root directory of program. | SmallElection1.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection1.csv” | SmallElection1.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 21:58:46 CST 2018\_audit.txt | 1 winner is chosen randomly. Both candidates get 1 vote. | 1 winner is chosen randomly. Both candidates get 1 vote. | This can be ran multiple times and the filename of audit file will change over time. |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi02** | **Name(s) of Testers: Shi Chen** |
| **Test Description: System testing for inputting an OPL file with 2 ballots but the candidates are written weirdly on the candidates’ line. To make sure the pattern matching is correct.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection2.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection2.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection2.csv from testing directory into root directory of program. | SmallElection2.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection2.csv” | SmallElection2.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 22:00:54 CST 2018\_audit.txt | 1 winner is chosen randomly. Both candidates get 1 vote. | 1 winner is chosen randomly. Both candidates get 1 vote. | This can be ran multiple times and the filename of audit file will change over time. |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi03** | **Name(s) of Testers: Shi Chen** |
| **Test Description: System testing for inputting an OPL file with 6 candidates, 3 parties, 9 ballots, and 3 seats. This is from the SRS specificiations document.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is OPL.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file OPL.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file OPL.csv from testing directory into root directory of program. | OPL.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “OPL.csv” | OPL.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 22:10:13 CST 2018\_audit.txt | Audit file with 3 winners. | Audit file with 3 winners. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi04** | **Name(s) of Testers: Shi Chen** |
| **Test Description: System testing for inputting an OPL file with 6 candidates, 3 parties, 9 ballots, and 3 seats. This is from the SRS specificiations document.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is OPL.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file OPL.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file OPL.csv from testing directory into root directory of program. | OPL.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “OPL.csv” | OPL.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 22:10:13 CST 2018\_audit.txt | Audit file with 3 winners. | Audit file with 3 winners. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi05** | **Name(s) of Testers: Shi Chen** |
| **Test Description: System testing for inputting an OPL file with 3 candidates, 3 parties, and 3-way tie for 2 seats. To check if ties are handled.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection3.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection3.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection3.csv from testing directory into root directory of program. | SmallElection3.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection3.csv” | SmallElection3.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 22:21:35 CST 2018\_audit.txt | Audit file with 2 winners. | Audit file with 2 winners. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi06** | **Name(s) of Testers: Shi Chen** |
| **Test Description: System testing for inputting an OPL file with 6 candidates, 2 parties, and 1 one party has twice as many votes but not enough to win 2 seats straight up, for 2 seats. To check if quota and remainder work properly.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection4.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection4.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection4.csv from testing directory into root directory of program. | SmallElection4.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection4.csv” | SmallElection4.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 22:24:06 CST 2018\_audit.txt | Audit file with 2 winners. | Audit file with 2 winners. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi07** | **Name(s) of Testers: Shi Chen** |
| **Test Description: System testing for inputting an OPL file with 6 candidates, 2 seats, 3 parties, a 3 way tie and to check if multiway with multiple candidates is handled properly.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection5.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection5.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection5.csv from testing directory into root directory of program. | SmallElection5.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection5.csv” | SmallElection5.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 22:28:10 CST 2018\_audit.txt | Audit file with 2 winners. | Audit file with 2 winners. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi07** | **Name(s) of Testers: Shi Chen** |
| **Test Description: System testing for inputting an OPL file with 6 candidates, 3 parties, 3 seats, a 3 way tie and to check if multiway with multiple candidates is handled properly with equal number of seats is properly handled (no party randomly jumps another for a seat, everyone has 1).** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection6.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection6.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection6.csv from testing directory into root directory of program. | SmallElection6.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection6.csv” | SmallElection6.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 22:30:06 CST 2018\_audit.txt | Audit file with 2 winners. | Audit file with 3 winners. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi08** | **Name(s) of Testers: Shi Chen** |
| **Test Description: System testing for inputting an OPL file with 7 candidates, 3 parties, uneven party numbers and 5 seats.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection7.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection7.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection7.csv from testing directory into root directory of program. | SmallElection7.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection7.csv” | SmallElection7.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 22:31:50 CST 2018\_audit.txt | Audit file with 5 winners. | Audit file with 5 winners. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi09** | **Name(s) of Testers: Shi Chen** |
| **Test Description: System testing for inputting an OPL file with 100000 ballots.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection8.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection8.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection8.csv from testing directory into root directory of program. | SmallElection8.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection8.csv” | SmallElection8.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 22:33:06 CST 2018\_audit.txt | Audit file with 5 winners, under 8 seconds. | Audit file with 5 winners, under 8 seconds. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi10** | **Name(s) of Testers: Shi Chen** |
| **Test Description: More seats than Candidates.** |  |
| **Automated: yes noX** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection9.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection9.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection9.csv from testing directory into root directory of program. | SmallElection9.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection9.csv” | SmallElection9.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 22:34:45 CST 2018\_audit.txt | Audit file with 11 winners, under 8 seconds. | Audit file with 7 winners, under 8 seconds. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Hong01** | **Name(s) of Testers: Qing Hong** |
| **Test Description: IR Voting with provided example file. Adjust observation to see some ballots were removed.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is IrTest1.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file IrTest1.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file IrTest1.csv from testing directory into root directory of program. | IrTest1.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “IrTest1.csv” | IrTest1.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | Election\_Thu Dec 06 22:41:07 CST 2018\_audit.txt  invalidated\_Thu Dec 06 22:41:07 CST 2018.txt | Audit file with correct winner. | Audit file with correct winner. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory. An invalidated txt file is written and the correct ballots have been removed.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi11** | **Name(s) of Testers: Shi Chen** |
| **Test Description: IR Voting with 100000 ballots.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection10.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection10.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection10.csv from testing directory into root directory of program. | SmallElection10.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection10.csv” | SmallElection10.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | invalidated\_Thu Dec 06 22:47:40 CST 2018.txt,  Election\_Thu Dec 06 22:47:40 CST 2018\_audit.txt | Audit file with correct winner. Complete in under 8 seconds. | Audit file with correct winner. Audit file with correct winner. Complete in under 8 seconds. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory. Table and invalidated file should also now be created.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi12** | **Name(s) of Testers: Shi Chen** |
| **Test Description: IR Voting with 2 ballots, 2 candidates, each getting a top rank. To test ties. Invalidated file should be blank.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection11.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection11.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection11.csv from testing directory into root directory of program. | SmallElection11.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection11.csv” | SmallElection11.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | invalidated\_Thu Dec 06 22:50:46 CST 2018.txt,  Election\_Thu Dec 06 22:50:46 CST 2018\_audit.txt | Audit file with correct winner. | Audit file with correct winner. Audit file with correct winner. | Random. Winner can be different. A winner just has to occur |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory. Blank invalidated file.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi13** | **Name(s) of Testers: Shi Chen** |
| **Test Description: IR Voting with 2 ballots, 2 candidates, each getting a top rank with weird names. To test file parsing.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection12.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail\_** |  |
|  |  |
| **Preconditions for Test:** file SmallElection12.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection12.csv from testing directory into root directory of program. | SmallElection12.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection12.csv” | SmallElection12.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | invalidated\_Thu Dec 06 22:58:17 CST 2018.txt,  Election\_Thu Dec 06 22:58:17 CST 2018\_audit.txt | Audit file with correct winner. | Audit file with correct winner. Audit file with correct winner. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory. Invalidated is blank.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi14** | **Name(s) of Testers: Shi Chen** |
| **Test Description: File will have ballot number inconsistent with the number of ballots listed and candidate count is also inconsistent with the number of candidates.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection13.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test:** file SmallElection13.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move the file SmallElection13.csv from testing directory into root directory of program. | SmallElection13.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection13.csv” | SmallElection13.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect audit and summary results for correctness. | invalidated\_Thu Dec 06 23:07:35 CST 2018.txt,  invalidated\_Thu Dec 06 23:14:08 CST 2018.txt | Should catch file inconsistencies. | No votes got distributed but a winner was declared. |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory. Invalidated is blank.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit \_ System X** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi15** | **Name(s) of Testers: Shi Chen** |
| **Test Description: File will have an elimination with a ballot whose #2 choice was also eliminated. Should still do so with invalids removed.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Data file is SmallElection14.csv is in the testing folder.  MyMenuDemo.java entry point is in src folder. |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test:** file SmallElection14.csv is in the root directory | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
|  | Move the file SmallElection14.csv from testing directory into root directory of program. | SmallElection14.csv | No errors | No errors |  |
| 2 | Compile all files in the src directory if they are not already compiled. Execute MyMenuDemo. Or run MyMenuDemo from Intellij |  | Compiles | Compiles |  |
| 3 | Enter the file name “SmallElection14.csv” | SmallElection14.csv | File is read with no errors. | File is read with no errors. |  |
| 4 | Inspect Tue Nov 13 15:19:26 CST 2018--[ SmallElection14.csv Audit].txt for correctness | invalidated\_Thu Dec 06 23:27:29 CST 2018.txt  Election\_Thu Dec 06 23:27:29 CST 2018\_audit.txt | File should get to another round of elections | File should get to another round of elections |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

An audit file named by time of test and the input file is in the directory. 2 ballots were removed in the invalidated file.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi16** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Ballot.getId(). It’s a standard test to make sure the getter returns the correct value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in BallotTest.java in function void getId(). |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute BallotTest | The test function tests an Id value of 132. | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi17** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Ballot.getVotes() with a small array with value on the front.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in BallotTest.java in function void getVotesSmallArrayFrontEdge(). |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute BallotTest | The test function tests an array with t[0] = 1. | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi18** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Ballot.getVotes() with a small array with value on the back.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in BallotTest.java in function void getVotesSmallArrayBackEdge(). |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute BallotTest | The test function tests an array with t[4] = 1. | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi19** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Ballot.getVotes() with a small array with value in the middle.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in BallotTest.java in function void getVotesSmallArrayMiddleEdge(). |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute BallotTest | The test function tests an array with t[2] = 1. | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi20** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Ballot.getVotes() with a big array with value on the front.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in BallotTest.java in function void void getVotesBigArrayFront() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute BallotTest | The test function tests an array with t[0] = 1. | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi21** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Ballot.getVotes() with a big array with value on the middle.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in BallotTest.java in function void getVotesBigArrayMiddle() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute BallotTest | The test function tests an array with t[50] = 1. | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi22** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Ballot.getVotes() with a big array with value on the back.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in BallotTest.java in function void getVotesBigArrayBack() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute BallotTest | The test function tests an array with t[99] = 1. | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi23** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Ballot.getCurRound()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in BallotTest.java in function void getCurRound() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute BallotTest | Test object begins with round of 1. | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi24** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Ballot.setCurRound(int)** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in BallotTest.java in function void setCurRound() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute BallotTest | Test object has round changed to 10. | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi25** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.getVotes()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void getVotesTest() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Test object has vote count of 10 | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi26** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.getVotes() on maximum value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void getVotesTestMax() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Test object tried to set votes to 9223372036854775808 | Assert test passes | Assert test fails |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi27** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.setVotes() on maximum value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void setVotes() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Test object tried to set votes to 9223372036854775808 | Assert test passes | Assert test fails |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi28** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.addVotes(int) on reasonable value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void addVotesNormal() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Test object tried to add 1 vote | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi29** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.addVotes(int) with normal high value of votes.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void addVotesHigh() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Test object tried to add 1 vote to 1000 votes | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi30** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Candidate.addVotes() on maximum value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void addVotesBadInput() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Test object tried to add 1 vote to 9223372036854775807 | Assert test passes | Assert test fails |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi26** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.getVotes() on maximum value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void getVotesTestMax() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Test object tried to set votes to 9223372036854775808 | Assert test passes | Assert test fails |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi31** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.Reset()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void Reset() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Test object tried to reset itself | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi32** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.** **addBallot()**  **on maximum value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void addBallot() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Add a Ballot object to a Candidate’s collection. | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi33** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.setBallots(ArrayList<Ballots>) on new list.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void setBallots() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Setting new ballots should change the ballots list and the vote count | Assert test passes | Assert test fails | Does not change votes count |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi34** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.getBallot()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void getBallots() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi35** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Candidate.toString() returns a string.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void toString() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi36** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.getParty()**  **After constructor.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void getPartyNull() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Candidate instance instantiated. | Assert test passes | Assert test fails | Candidate’s party is set to null on certain constructors. |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi37** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Candidate.setParty()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void setParty() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi38** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Candidate.setName()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void voidNameTest() |
| **Results: Pass Fail x** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Pass a null to setName() | Assert test passes | Assert test fails | Does not catch null parameters. |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi39** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Candidate.setParty()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void voidParty() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Add a null object to a Candidate’s collection. | Assert test passes | Assert test passes | Should not have passed. Should never pass a null to collection. |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi40** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.getId()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void getId() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi41** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.getStatus()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void getStatus() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi42** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.getName()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void getName() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi43** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.setId(long)** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void setId() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi44** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.setName(String)** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void setName() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi45** | **Name(s) of Testers: Shi Chen** |
| **Test Description: TestCandidate.setStatus(int)** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void setStatus() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi46** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Candidate.CompareTo(Candidate)** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void compareToEquals() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Another Candidate instance | Assert test passes | Assert test fails | Randomly breaks ties so it should occasionally fail. |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi47** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Candidate.CompareTo(Candidate)** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in CandidateTest.java in function void compareToBadValue() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute CandidateTest | Another Candidate instance with bad vote count (pass limit for long) | Assert test passes | Assert test fails | Does not check negative values |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi48** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.getTotalCounts ()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void getTotalCounts() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi49** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.setTotalCounts (long)** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void setTotalCounts() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest | 9223372036854775807L +1 | Assert test passes | Assert test fails | Does not handle oversized longs. |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi50** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.setSeatCounts (long)** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void setSeatCounts() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest | 9223372036854775807L +1 | Assert test passes | Assert test fails | Does not handle oversized longs. |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi51** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.getSeatCounts()** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void getSeatCounts() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi52** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.compareTo(Party) with overflowed value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void compareToEqualsBadValue() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test fails | Does not handle overflow |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi53** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.compareTo(Party) with equal value on vote count.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void compareToEquals() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test fails | Fails occasionally because the value returned is random. |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi54** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.addSeats(1) with overflowed value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void addSeats() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test fails | Does not handle overflow |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi55** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.addTotalCounts(1) with overflowed value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void addTotalCounts() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test fails | Does not handle overflow |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi56** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.setCandidates(Candidate) with null value** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void setCandidates() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test fails | Does not handle null values for setter. |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi57** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.getCandidates() after adding a Candidate.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void getCandidates() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test fails | Does not handle null values for setter. |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi58** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.addCandidate() will null value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void addCandidate() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test fails | Does not handle null values for adding candidate. |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi59** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.toString() returns a non empty.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void toStringTest() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi60** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.getName() returns a non empty.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void getName() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test passes |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi61** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Party.setName(String) will null value.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in PartyTest.java in function void setName() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Navigate to the src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute PartyTest |  | Assert test passes | Assert test fails | Does not handle null names. |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi62** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Util.getFirstLine(String) with first line not “OPL” or “IR”** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in UtilTest.java in function void getFirstLine () |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move auditrand3.csv from testing folder to src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute UtilTest | Auditrand3.csv | Assert test passes | Assert test fails | Does not handle files where first line is not OPL or IR |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi63** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test Util.getFirstLine(String) with first line “OPL” or “IR”** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in UtilTest.java in function void getFirstLine Normal() |
| **Results: Pass Fail X** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move auditrand2.csv from testing folder to src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute UtilTest | Auditrand2.csv | First line read was “OPL” | First line read was “OPL” |  |
|  |  |  | . |  |  |
|  |  |  |  |  |  |

**Post condition(s) for Test:**

None. Test either passes or fails.

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Hong02** | **Name(s) of Testers: Qing Hong** |
| **Test Description: Test IRVoting.parseInputFile(String) to see if the correct number of candidates and ballots were read.** |  |
| **Automated: yes X no** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in IRVotingTest.java in function void testParseInputFileIR() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move irTest1.csv from testing folder to src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute IRVotingTest | irTest1.csv | 6 ballots and 4 candidates were read. | 6 ballots and 4 candidates were read. |  |
|  |  |  | . |  |  |

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Hong03** | **Name(s) of Testers: Qing Hong** |
| **Test Description: Test IRVoting.run() to see if the correct winner was found.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in IRVotingTest.java in function void testRun() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move irTest1.csv from testing folder to src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute IRVotingTest | irTest1.csv | Candidate Rosen was the winner | Candidate Rosen was the winner |  |
|  |  |  | . |  |  |

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi64** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test OPLVoting.parseInputFile() to see if file was read with no error.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in OPLVoting.java in the main() to access the private parseInputFile(String) function. |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move SmallElection1.csv from testing folder to src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute OPLVoting | SmallElection1.csv | File read with no error | File read with no error |  |
|  |  |  | . |  |  |

|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 3** | |
| **Test Stage: Unit X System** | **Test Date: 12/6/2018** |
| **Test Case ID#: TC-Shi65** | **Name(s) of Testers: Shi Chen** |
| **Test Description: Test OPLVoting.run() to see if the correct winner was found.** |  |
| **Automated: yes no X** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  Test is stored in OPLVoting.java in function void main() |
| **Results: Pass X Fail** |  |
|  |  |
| **Preconditions for Test: You are in the src directory.** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Move SmallElection3.csv from testing folder to src folder |  |  |  |  |
| 2 | Compile all the .java files |  | Compiles | Compiles |  |
| 3 | Execute OPLVoting | SmallElection3.csv | File read with no error | File read with no error |  |
| 4 | Inspect audit file for correctness |  | 2 biggest parties came out with seats. | 2 biggest parties came out with seats. |  |