|  |  |
| --- | --- |
| **Project Name: Aegis Project 1: Voting System Team#7** | |
| **Test Stage: Unit X System \_** | **Test Date:** 3/26/21 |
| **Test Case ID#:** IRTest\_2 | **Name(s) of Testers:** Michael/Lucky/Grant/Donald |
| **Test Description:**  TEST\_F (IRTest, findWinnerNoWinnernoTie)  Checks for functionality of findWinners  Where there is a winner or a tie. |  |
| **Automated: yes X no \_\_\_** | **Indicate where you are storing the tests (what file) and the name of the method/functions being used.**  Stored in file: IRTest.cpp  Method names:   * findWinner() |
| **Results: Pass \_\_X\_\_\_** Fail **\_\_\_\_\_\_\_\_** |  |
|  |  |
| **Preconditions for Test:**  The data for the IR Election was set up correctly. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
|  |  |  |  |  |  |
| 1 | Set up the first candidate | &ir->getCandidates()[0] |  |  |  |
| 2 | Check if the candidate has a majority. | float temp = (float)ir->getElim()->getNumBallots() / 6.0; | EXPECT\_LE(temp, 0.5) = false | EXPECT\_LE(temp, 0.5) = false |  |
|  |  |  |  |  |  |

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

The candidate should have a majority.