|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team#3** | |
| **Test Stage: Unit \_x\_ System \_\_** | **Test Date: 4/1/20** |
| **Test Case ID#: Plurality\_election\_record\_UT003** | **Name(s) of Testers: Colin Kluegel** |
| **Test Description:**  **Test distributes the ballots to candidates and then sorts the list by number of ballots, test verifies that candidates are correctly sorted** | **Test file: plurality\_election\_record\_UT.cc**  **Method: TEST\_F(PluralityElectionRecordTests, SortNonElectedCandidateList)** |
| **Automated: yes\_x\_\_ no \_\_\_** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.** |
| **Results: Pass \_\_\_x\_\_ Fail\_\_\_\_\_\_\_\_** |  |
|  |  |
| **Preconditions for Test: 5 candidate objects and 5 ballot objects are created and put in candidate lists and ballot lists respectively. A new PluralityElectionRecord object is created with these lists** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| 1 | Call election\_record->DistributeBallots |  |  |  |  |
| 2 | Call election\_record->SortNonelectedCandidateList |  |  |  |  |
| 3 | Create a list of candidates and set it to what is returned from election\_record->GetNonElectedCandidateList() | Candidate list |  |  |  |
| 4 | Check that candidate at front of the list is the correct winner | Candidate list | Candidate with ID 1 at front of list | Candidate with id 1 at front of list  Test passed |  |
| 5 | Remove first candidate from list so we can check 2nd place is correct | Candidate on list | Candidate with ID 2 at front of list | Candidate with id 2 at front of list  Test passed |  |
|  |  |  |  |  |  |
| 6 |  |  |  |  |  |

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

All ballots have been distributed, nonElectedCandidateList is now sorted with the candidates with more votes at the front of the list.

Project Name: The project #, name of your system, and the team#

Test Stage: Indicate whether it is a unit test or a system test.

Test Date: The date the test was performed.

Test Case ID#: A unique ID is required. Decide on a naming convention and use numbering. Example: Ballot\_Shuffle\_1

Name(s) of Testers: List the names of anyone involved in running this test case.

Test Description: Describe briefly the test objective.

Automated: Indicate if the test is completely automated or being checked manually. (If you have methods running the tests and checking results, select “yes”. If you are manually checking results, indicate manual by selecting the “no.”)

**Results:** Indicate if the test passed or failed.

**Step #:** You will be listing the test steps in order. This number is the step number in the process.

**Test Step Description:** Details of the test step.

**Test Data:** What the test data will be for this step. Be clear on what the input data will be. If using a specific file, be clear on the name.

**Expected Result:** What result are you expecting from the program component or system.

**Actual Result:** What result were returned based on the test.

**Post condition for Test:** What will be true after the test has been run? Has the state of the system changed in any way?

**Notes:** Comments and notesfor you and your team members.