|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team#3** | |
| **Test Stage: Unit \_x\_ System \_\_** | **Test Date: 4/1/20** |
| **Test Case ID#: Plurality\_election\_record\_UT009** | **Name(s) of Testers: Colin Kluegel** |
| **Test Description:**  **Checks that when candidates are moved to the losers list all the non-elected candidates are successfully moved to the losers list** | **Test file: plurality\_election\_record\_UT.cc**  **Method: TEST\_F(PluralityElectionRecordTests, GetLosersList)** |
| **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: 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 | Check that losers list is empty |  | Losers list is empty | Losers list is empty, test passes |  |
| 2 | Move 2 candidates to winners list |  |  |  |  |
| 3 | Move the rest of the candidates to the loser list |  |  |  |  |
| 4 | Create a loser\_list of candidates from what is returned by election\_record->MoveRemainingCandidatesToLosersList() | Loser\_list |  |  |  |
| 5 | Check size of losers list |  | Size is 3 | Size is 3, test passes |  |
|  |  |  |  |  |  |
| 6 | Check that the correct candidate is at the front of the losers lsit |  | Candidate3 is at the front of the losers list | Candidate3 is at the front of the losers list, test passed |  |
|  |  |  |  |  |  |
| 7 | Pop first candidate off of loser\_list, |  |  |  |  |
| 8 | Create a candidate losing\_candidate that is equal to the first element in loser\_list | Losing\_candidate |  |  |  |
| 9 | Check that losing\_candidate is the correct candidate |  | Candidate2 is the losing\_candidate | Candidate2 is the losing\_candidate, test passed |  |
|  |  |  |  |  |  |
| 10 | Pop candidate off of loser\_list |  |  |  |  |
| 11 | Set losing\_candidate to front of loser list | Losing\_candidate |  |  |  |
| 12 | Check that losing\_candidate is the correct candidate |  | Candidate1 is the losing\_candidate | Candidate1 is the losing\_candidate, test passed |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

2 candidates are on the winners list and 3 are on the losers 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.