| **Project Name: Project 1: Voting System Team# 9** | |
| --- | --- |
| **Test Stage: Unit YES** | **Test Date: 03/26/2021** |
| **Test Case ID#: testGetNameLowerCase()** | **Name(s) of Testers: Isaac Xiong** |
| **Test Description:**  Name should be case-sensitive, so that means that lower-case and upper-case should not be returned or expected.  Tests lower-case. |  |
| **Automated: Yes** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  The file used to test the Candidate class is CandidateTest.java. These are the methods being tested: getName() |
| **Results: Pass** |  |
|  |  |
| **Preconditions for Test:**  A Candidate object must be created. It may or may not contain ballots in it’s ArrayList<Ballot> member variable. | |

| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Gets the name of the candidate and checks if it is correct.”terry” is not equal to “Terry”, and therefore return true in the assertThat(). | Candidate cand1 = new Candidate("Terry", 'R');  int[] choices1 = new int[] {1,0,0};  int[] choices2 = new int[] {1,2,3};  Ballot b1 = new Ballot(1,choices1);  Ballot b2 = new Ballot(2, choices2); | “terry” | “Terry” | “terry” is not equal to “Terry”, therefore passing this test |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

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

The Candidate will been created with the correct name.

**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.