Project Name: Project 2: Voting System **Team# 09**

Test Stage: Unit _✔_ System __ **Test Date:** April 22 2018

Test Case ID#: BT_01 **Name(s) of Testers:** Jay Uppaluri

Test Description: Testing the default values in the ballot object constructor and the getters/setters.

The unit tests are in the unittest.cc in the testing folder. Once user runs make, a directory is created in the testing folder. Traverse the directory (/build/bin) and execute the executable (unittest).

Automated: yes_✔_ no ___

Results: Pass _✔__ Fail_____

Preconditions for Test:

1. Default values are set in the constructor
    a. if the default values in the constructor are altered, the unittest must be compiled again
    b. expected values are fixed constants

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Instantiate the class object	Default values of the default constructor in ballot.cc	ballot_id = 5 num_candidates = 0 list_of_names_ = NULL list_of_ranks_ = NULL	ballot_id = 5 num_candidates = 0 list_of_names_ = NULL list_of_ranks_ = NULL	
2	Calling the object's setters	ballot_id = 10 num_candidates = 5 list_of_names_ = {"john", "mary", "bromeo"} list_of_ranks_ = {1, 2, 3, 4}	ballot_id = 10 num_candidates = 5 list_of_names_ = {"john", "mary", "bromeo"} list_of_ranks_ = {1, 2, 3, 4}	ballot_id = 10 num_candidates = 5 list_of_names_ = {"john", "mary", "bromeo"} list_of_ranks_ = {1, 2, 3, 4}	

Post condition(s) for Test: The object's variables are set to the values in step 2 due to the setters. After testing, the object is destroyed via the destructor.

Project Name: Project 2: Voting System — Team# 09

Test Stage: Unit _✔_ System __ **Test Date:** April 22 2018

Test Case ID#: BT_02 **Name(s) of Testers:** Jay Uppaluri

Test Description: Testing the ballot.toString function

The unit tests are in the unittest.cc in the testing folder. Once user runs make, a directory is created in the testing folder. Traverse the directory (/build/bin) and execute the executable (unittest).

Automated: yes_✔_ no ___

Results: Pass _✔__ Fail_____

Preconditions for Test:

1. ballot_id exists, num_candidates exists, and list_of_ranks exists

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Instantiate the class object	Default values of the default constructor in ballot.cc	ballot_id = 5 num_candidates = 0 list_of_names_ = NULL list_of_ranks_ = NULL	ballot_id = 5 num_candidates = 0 list_of_names_ = NULL list_of_ranks_ = NULL	
2	Calling the object's setters	ballot_id = 10 num_candidates = 4 ranks = {1, 2, 3, 4}	ballot_id = 10 num_candidates = 4 ranks = {1, 2, 3, 4}	ballot_id = 10 num_candidates = 4 ranks = {1, 2, 3, 4}	

3	Test toString	ballot_id = 10 num_candidates = 4 ranks = {1, 2, 3, 4}	ballot.toString() = "ID: 10 Count: 4\n1 2 3 4"	ballot.toString() = "ID: 10 Count: 4\n1 2 3 4"	

Post condition(s) for Test:

toString() returns the expected result in step 3. After testing, the object is destroyed via destructor and therefore, the system state remains the same.

Project Name: Project 2: Voting System — Team# 09

Test Stage: Unit _✔_ System __ **Test Date:** April 22 2018

Test Case ID#: BT_03 **Name(s) of Testers:** Jay Uppaluri

Test Description: Testing the ballot.findCandidate function.

The unit tests are in the unittest.cc in the testing folder. Once user runs make, a directory is created in the testing folder. Traverse the directory (/build/bin) and execute the executable (unittest).

Automated: yes_✔_ no ___

Results: Pass _✔__ Fail_____

Preconditions for Test:

2. num_candidates exists, ranks is an array and exists.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Instantiate the class object	Default values of the default constructor in ballot.cc	ballot_id = 5 num_candidates = 0 list_of_names_ = NULL	ballot_id = 5 num_candidates = 0 list_of_names_ = NULL	

			list_of_ranks_ = NULL	list_of_ranks_ = NULL	
2	Calling the object's setters	num_candidates = 5 ranks = {1, 2, 3, 4}	num_candidates = 5 ranks = {1, 2, 3, 4}	num_candidates = 5 ranks = {1, 2, 3, 4}	
3	Test findCandidate	num_candidates = 5 ranks = {1, 2, 3, 4}	ballot.findCandidate(1) = 0;	ballot.findCandidate(1) = 0;	

Post condition(s) for Test:

The findCandidate function returns the index of the candidate specified. After testing, the object is destroyed via destructor and therefore, the system state remains the same.

Project Name: Project 2: Voting System — Team# 09

Test Stage: Unit ✔ System __ **Test Date: April 22 2018**

Test Case ID#: Ballot_SetListOfRank_04 **Name(s) of Testers:** Shalom Nguyen

Test Description: To test the setter method for the array containing each candidate's rankings.

The unit tests are in the unittest.cc in the testing folder. Once user runs make, a directory is created in the testing folder. Traverse the directory (/build/bin) and execute the executable (unittest).

Automated: yes✔ no ___

Results: Pass ✔ Fail___

Preconditions for Test:

an array containing a rank per candidate

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	instantiate ballot object	Default values of the default constructor in ballot.cc	ballot_id = 5 num_candidates = 0 list_of_names_ = NULL	ballot_id = 5 num_candidates = 0 list_of_names_ = NULL	

			list_of_ranks_ = NULL	list_of_ranks_ = NULL	
2	call the setBallot_id() method	ballot_id = 10	ballot_id = 10	ballot_id = 10	
3	call the setNum_candidates() method	num_candidates = 5	num_candidates = 5	num_candidates = 5	

4	call the setList_of_ranks()	a[0] = 0;
		a[1] = 1;
		a[2] = 2;
		a[3] = 3;
			a[0] = 0;
			a[1] = 1;
			a[2] = 2;
			a[3] = 3;
				a[0] = 0;
				a[1] = 1;
				a[2] = 2;
				a[3] = 3;
					The precondition for test and allocate an array (i.e. int* a = new int[4];)

5	Test setList_of_ranks()	a[0] = 0;
		a[1] = 1;
		a[2] = 2;
		a[3] = 3;
			a[0] = 0;
			a[1] = 1;
			a[2] = 2;
			a[3] = 3;
				a[0] = 0;
				a[1] = 1;
				a[2] = 2;
				a[3] = 3;
					Test via getList_of_ranks with 'a' (see (4))

Post condition(s) for Test: SetListofRank set the array containing each candidate's rankings to what the values in step 4. In its current state, the object, in this case, ballot, now has been assigned the above values (5). However, after testing, the object is destroyed via destructor and therefore, the system state remains the same.