

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: ballot_test_01	Name(s) of Testers: Nicholas Lovdahl
Test Description: When a ballot is created, its ID should be the same as the ID provided to it when it was made. This test creates ballots with random IDs and calls get_id() to see that it returns the same ID.	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "Ballot.java". The class with this test, "test_get_id()", is in "BallotTest.java"
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: N/A

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Generate a random number.				
2	Create a ballot with 'null' for rankings and the random number from step 1 for the ID.				
3	Call 'get_id()'.		The number returned is the same as the number from step 1.	The number returned is the same as the number from step 1.	
4	Repeat steps 1-3 9 more times.				

Postconditions for Test: The ID returned was the same as the ID given for construction for each ballot.
--

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: ballot_test_02	Name(s) of Testers: Nicholas Lovdahl
Test Description: When a ballot is created, its current rank attribute should initially be 0. When 'current_rank()' is called immediately after this, 0 should be returned.	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "Ballot.java". The class with this test, "test_starting_rank()", is in "BallotTest.java"
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: N/A

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a ballot with 'null' for rankings and 101 for its ID.				
2	Call 'current_rank()'.		0 should be returned.	0 was returned.	

Postconditions for Test: N/A

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: ballot_test_03	Name(s) of Testers: Nicholas Lovdahl
Test Description: When 'increment_rank()' is called, it should increase the current rank attribute for a ballot and return that value. It should only increment by one at a time, and it should return -1 if the rank is incremented more times than there are rankings.	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "Ballot.java". The class with this test, "test_ending_rank()", is in "BallotTest.java"
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: N/A

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a ballot with an array for 5 candidates (an array of 5 nulls) for its rankings and 101 for its ID.				
2	Increment the rank 4 times and check that each time, an incremented value is returned (1, 2, 3, and finally 4).		1, 2, 3, and finally 4 are returned, respectively.	1, 2, 3, and 4 were returned.	
3	Increment the rank one more time. -1 should be returned since we have an array for 5 rankings.		-1 is returned.	-1 was returned.	
4	Increment the rank two more times. -1 should be returned each time.		-1 is returned each time.	-1 was returned each time.	

Postconditions for Test: Incrementing the rank will return -1.

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: ballot_test_04	Name(s) of Testers: Nicholas Lovdahl
Test Description: Incrementing a ballot more times than there are rankings should never advance a ballots current rank attribute to a number greater than the number of rankings it has. Even when increment rank is called multiple times afterwards, 'current_rank()' should still return the greatest index for the rankings, but never more than that.	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "Ballot.java". The class with this test, "test_ending_rank()", is in "BallotTest.java"
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: N/A

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a ballot with an array for 5 candidates (an array of 5 nulls) for its rankings and 101 for its ID.				
2	Increment the rank 4 times.				
3	Call 'current_rank()' and check that the value returned is 4.		The value returned should be 4.	The value returned was 4.	
4	Increment the rank one more time.				
5	Call 'current_rank()' and check that the value returned is 4.		The value returned should still be 4.	The value returned was 4.	
6	Increment the rank three more times.				
7	Call 'current_rank()' and check that the value returned is 4.		The value returned should still be 4.	The value returned was 4.	

Postconditions for Test: Incrementing the rank and then checking the current rank for the ballot will be maxed out at 4.

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: ballot_test_05	Name(s) of Testers: Nicholas Lovdahl
Test Description: It is important that 'current_rank()' and 'increment_current_rank()' work in tandem. Getting the current rank should not increment it and incrementing the current rank should only increment the rank one at a time without ever exceeding the range of rankings. This test checks that these two methods work together and behave as expected.	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "Ballot.java". The class with this test, "test_rank_advancement()", is in "BallotTest.java"
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: N/A

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a ballot with an array for 5 candidates (an array of 5 nulls) for its rankings and 101 for its ID.				
2	Call 'current_rank()' three times. Check that it returns 0 each time (the rank should not advance).		The value returned should be 0 each time.	The value returned was 0 each time.	
3	Call 'increment_rank()' once.				
4	Call 'current_rank()' once. Check that it returns 1.		The value returned should be 1.	The value returned was 1.	
5	Call 'current_rank()' three times. Check that it returns 1 each time (the rank should not advance).		The value returned should be 1 each time.	The value returned was 1 each time.	
6	Call 'increment_rank()' three times.				

Straightforward Voting System - Testing Document

Team #2

7	Call 'current_rank()' once. Check that it returns 4.		The value returned should be 4.	The value returned was 4.	
---	--	--	---------------------------------	---------------------------	--

Postconditions for Test: The current rank of the ballot is 4.

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: ballot_test_06	Name(s) of Testers: Nicholas Lovdahl
Test Description: This test checks to see that calling 'reset_current_rank()' sets the current rank of a ballot to zero, no matter what the current state of the rank is.	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "Ballot.java". The class with this test, "test_reset_current_rank()", is in "BallotTest.java"
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: N/A

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create a ballot with an array for 5 candidates (an array of 5 nulls) for its rankings and 101 for its ID.				
2	Call 'reset_current_rank()'.				
3	Call 'current_rank()' once. Check that it returns 0.		The value returned is 0.	The value returned was 0.	
4	Call 'increment_rank()' 4 times.				
5	Call 'reset_current_rank()'.				
6	Call 'current_rank()' once. Check that it returns 0.		The value returned is 0.	The value returned was 0.	
7	Call 'increment_rank()' 6 times.				
8	Call 'reset_current_rank()'.				

Straightforward Voting System - Testing Document

Team #2

9	Call 'current_rank()' once. Check that it returns 0.		The value returned is 0.	The value returned was 0.	
---	--	--	--------------------------	---------------------------	--

Postconditions for Test: The current rank of the ballot is 0.

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: ballot_test_07	Name(s) of Testers: Nicholas Lovdahl
Test Description: Ballots need to be able to provide a reference to the candidate currently chosen	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "Ballot.java". The class with this test, "test_current_choice()", is in "BallotTest.java". This test also makes use of the constructor for the Candidate class in "Candidate.java".
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: The constructor for the Candidate class works and does not produce errors.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Create an array of 5 distinct candidates. The candidates can be named with letters A-E and IDs 0-4.				
2	Create a ballot with the array of candidates for rankings and 101 for the ID.				
3	Call 'current_choice()' twice. Check that the candidate returned is the same (not just a copy) as the first candidate created.		The candidate returned should be the first candidate each time.	The first candidate was returned each time.	
4	Call 'increment_rank()' once.				
5	Call 'current_choice()' once. Check that the candidate returned is the same as the second candidate.		The candidate returned should be the second candidate.	The second candidate was returned.	

Straightforward Voting System - Testing Document

Team #2

6	Call 'increment_rank()' once.				
7	Call 'current_choice()' once. Check that the candidate returned is the same as the third candidate.		The candidate returned should be the third candidate.	The third candidate was returned.	
8	Call 'increment_rank()' once.				
9	Call 'current_choice()' once. Check that the candidate returned is the same as the fourth candidate.		The candidate returned should be the fourth candidate.	The fourth candidate was returned.	
10	Call 'increment_rank()' once.				
11	Call 'current_choice()' twice. Check that the candidate returned is the same as the fifth candidate created.		The candidate returned should be the fifth candidate each time.	The fifth candidate was returned each time.	

Postconditions for Test: The candidate returned by calling 'current_choice()' will be the fifth candidate.

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): System	Test Date: 4/2/2020
Test Case ID#: system_stress_test_01	Name(s) of Testers: Nicholas Lovdahl
Test Description: One of the non-functional requirements for SVS is that it is able to determine the results of an election within 5 minutes. SVS should be able to fulfill this requirement up to 100,000 ballots and 10 candidates (the maximum per specifications). This test determines whether SVS can determine the results of a plurality type election within this time frame when operating with the upper bounds of what it was designed for (in terms of ballots and candidates).	Indicate where you are storing the tests (what file) and the name of the method being used: This test will use "stress_test.csv" as testing data. This file contains 100,000 ballots for 10 candidates.
Automated (yes or no): No	
Results (pass or fail): Pass	

Preconditions for Test: SVS can be compiled, load files, and run elections without errors, and the file with the testing data is ready to be read.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Start the Straightforward Voting System.				
2	Set the number of seats to 10 and the election type to 'Plurality'.				
3	Start a timer on a separate device.				
4	Load the file with the testing data.	stress_test.csv			
5	Click on the button to run an election. Stop the timer once the results of that election are returned. Record the time needed as given by the timer.		The time recorded by the timer should be any amount of time less than 5 minutes.	The time required was about 10 seconds.	This measurement may not be as accurate due to the time required to stop / start the timer relative to the time taken by the program (too fast for me).

Postconditions for Test: SVS has determined the results for a plurality election with 10 candidates and 100,000 ballots from the file of testing data.

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): System	Test Date: 4/2/2020
Test Case ID#: system_stress_test_02	Name(s) of Testers: Nicholas Lovdahl
Test Description: One of the non-functional requirements for SVS is that it is able to determine the results of an election within 5 minutes. SVS should be able to fulfill this requirement up to 100,000 ballots and 10 candidates (the maximum per specifications). This test determines whether SVS can determine the results of a STV with Droop quota type election within this time frame when operating with the upper bounds of what it was designed for (in terms of ballots and candidates).	Indicate where you are storing the tests (what file) and the name of the method being used: This test will use "stress_test.csv" as testing data. This file contains 100,000 ballots for 10 candidates.
Automated (yes or no): No	
Results (pass or fail): Pass	

Preconditions for Test: SVS can be compiled, load files, and run elections without errors, and the file with the testing data is ready to be read.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Start the Straightforward Voting System.				
2	Set the number of seats to 10 and the election type to 'STV with Droop Quota'.				
3	Start a timer on a separate device.				
4	Load the file with the testing data.	stress_test.csv			
5	Click on the button to run an election. Stop the timer once the results of that election are returned. Record the time needed as given by the timer.		The time recorded by the timer should be any amount of time less than 5 minutes.	The time required was about 10 seconds.	This measurement may not be as accurate due to the time required to stop / start the timer relative to the time taken by the program (too fast for me).

Postconditions for Test: SVS has determined the results for a STV with Droop quota election with 10 candidates and 100,000 ballots from the file of testing data.

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: Election_controller_01	Name(s) of Testers: Arun Sharma
Test Description: The ballots get distributed among candidates, it should update the correct number of addition inside a particular candidate.	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "ElectionController.java". The class with this test, "distributeBallot()", is in "ElectionControllerTest.java"
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: N/A

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Read ballots from the testing folder and save it into the arraylist.	test.csv			
2	Add 1st ballot to a dummy candidate and compare with Integer 1 (AssertEqual)	test.csv	Candidate Size = 1	Candidate Size = 1	
3	Add 2nd t ballot to a dummy candidate and compare with Integer 2 (AssertEqual)	test.csv	Candidate Size = 2	Candidate Size = 2	
4	Add 3rd ballot to a dummy candidate and compare with Integer 3 (AssertEqual)	test.csv	Candidate Size = 3	Candidate Size = 3	
5	Add 4th ballot to a dummy candidate and compare with Integer 4 (AssertEqual)	test.csv	Candidate Size = 4	Candidate Size = 4	

Postconditions for Test: The Arraylist will return the same number of ballots assigned to the candidates as the expected in the output.
--

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: Election_controller_02	Name(s) of Testers: Arun Sharma
Test Description: The ballots get redistributed among candidates, it should update or remove the correct number of addition inside a particular candidate.	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "ElectionController.java". The class with this test, "redistributeBallot()", is in "ElectionControllerTest.java"
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: N/A

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Read ballots from the testing folder and save it into the arraylist.	test.csv			
2	Add 1st ballot to a dummy candidate and compare with Integer 1 (AssertEqual)	test.csv	Candidate Size = 1 (true)	Candidate Size = 1	
3	Add and remove 2nd ballot to a dummy candidate and compare with Integer 1 (AssertEqual)	test.csv	Candidate Size = 1 (true)	Candidate Size = 1	
4	Add 3rd ballot from the dummy candidate and compare with Integer 1 (AssertEqual)	test.csv	Candidate Size = 2 (true)	Candidate Size = 2	
5	Add and remove the 4th ballot to a dummy candidate and compare with Integer 2 (AssertEqual)	test.csv	Candidate Size = 2 (true)	Candidate Size = 2	

Postconditions for Test: The Arraylist will return the same number of ballots assigned to the candidates as the expected in the output.
--

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: Election_controller_03	Name(s) of Testers: Arun Sharma
Test Description: When the ballots get distributed or redistributed among candidates, record every message and provide the correct number of messages recorded.	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "ElectionController.java". The class with this test, "customAuditFileMsg()", is in "ElectionControllerTest.java"
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: N/A

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Read ballots from the testing folder and save it into the arraylist.	test.csv			
2	Add first message to the file and check the message size if it's equal to 1 (AssertEqual)	test.csv	Message Size = 1	Message Size = 1	
3	Add first message to the file and check the message size if it's equal to 2 (AssertEqual)	test.csv	Message Size = 2	Message Size = 2	
4	Add first message to the file and check the message size if it's equal to 3 (AssertEqual)	test.csv	Message Size = 3	Message Size = 3	
5	Add first message to the file and check the message size if it's equal to 4 (AssertEqual)	test.csv	Message Size = 4	Message Size = 4	

Postconditions for Test: The Arraylist will return the same set of length as the expected output.
--

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: Ballot_Factory_01	Name(s) of Testers: Arun Sharma
Test Description: The ballots get stored in an array list. The array list should have the correct number of entries, correct ID of the given ballots etc.	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "BallotFactory.java". The class with this test, "initBallots()", is in "BallotFactoryTest.java"
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: N/A

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Read ballots from the testing folder and save it into the arraylist.	test.csv			
2	Check the total number of ballots and compare with Integer 30.	test.csv	30 (true)	30 (true)	
3	Check the ID of the first element if the total ballots are not shuffled and compare it to 0.	test.csv	0 (true)	0 (true)	
4	Check the name of the first element if the total ballots are not shuffled and compare it to A.	test.csv	A (true)	A (true)	

Postconditions for Test: The arraylist will return the same set of input as expected providing correctness of the method.
--

Straightforward Voting System - Testing Document

Team #2

Test Stage (Unit or System): Unit	Test Date: 4/2/2020
Test Case ID#: Ballot_Factory_02	Name(s) of Testers: Arun Sharma
Test Description: The ballots bet's distributed among candidates, it should update the correct number of candidates	Indicate where you are storing the tests (what file) and the name of the method being used: The class being tested is in "ElectionController.java". The class with this test, "initCandidates()", is in "BallotFactoryTest.java"
Automated (yes or no): Yes	
Results (pass or fail): Pass	

Preconditions for Test: N/A

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Read ballots from the testing folder and save it into the arraylist.	test.csv			
2	Check the 1st candidate name and compare it to A	test.csv	A(true)	A(true)	
3	Check the 2nd candidate name and compare it to B	test.csv	B(true)	B(true)	
4	Check the 3rd candidate name and compare it to C	test.csv	C(true)	C(true)	
5	Check the 4th candidate name and compare it to D	test.csv	D(true)	D(true)	
6	Check the 5th candidate name and compare it to E	test.csv	E(true)	E(true)	
7	Check the total candidate length	test.csv	6 (true)	6 (true)	

Postconditions for Test: The arraylist will return the same set of input as expected.
--