Project Name: VotingSystem
Testers: Matthew Johnson, Bek Allenson, Logan Watters, Perrie Gryniewicz

Test Case #	Test Case Description	Test Data	Preconditions	Steps to execute	Actual result	Expected Result	Result (pass/fail)	Comment
tc_001	testCheckMajorityTrue	IRVotingSystemTest.java	Junit is set up	Run junit IRVotingSystemTest.java	true	true	pass	
tc_002	testCheckMajorityFalse	IRVotingSystemTest.java	Junit is set up	Run junit IRVotingSystemTest.java	false	false	pass	
tc_003	testBreakTie	IRVotingSystemTest.java	Junit is set up	Run junit IRVotingSystemTest.java	True, True, True, True	True, True, True, True	pass	
tc_004	testBreakTieMultiple	IRVotingSystemTest.java	Junit is set up	Run junit IRVotingSystemTest.java	True, True, True	True, True, True	pass	
tc_005	testConstructor	IRBallotTest.java	Junit is set up	Run junit IRBallotTest.java	Test passed	Test passed	pass	
tc_007	testGetCandidate	IRBallotTest.java	Junit is set up	Run junit IRBallotTest.java	Test passed	Test passed	pass	
tc_008	testGetNextCandidate	IRBallotTest.java	Junit is set up	Run junit IRBallotTest.java	Test passed	Test passed	pass	

tc_009	testParseHeaderIR	HeaderprocessorTest.java	Junit is set up	Run junit HeaderProcessorTest.java	Test passed	Test passed	pass	
tc _010	An IRElection where no majority is found after all ballots have been redistributed	electionTest1.csv	Election file contains ballots and will lead to a tie in the middle of the election	Run program, pass in “electionTest1.csv” when prompted	expected	If Rosen is removed, Chou wins. If Kleinberg is removed, Rosen wins. If Chou is removed, Kleinberg wins	pass	
tc_011	An IR Election where no majority is found after all ballots have been redistributed so the candidate with the most ballots is selected as the winner	electionTest3.csv	Election file contains ballots and after all redistributions there is no majority winner	Run program, pass in “electionTest3.csv” when prompted	expected	Rosen is declared the winner with 4/10 votes	pass	
tc_012	Test the OPL system with 0 ballots	OPLNoBallot.csv	No ballots are given to the FileProcessor	Run program, pass in “OPLNoBallot.csv”	DivideByZero error	Error	failed	
tc_013	Test the OPL system with ballots	OPLElectionTest.csv	Ballots have been created	Run program, pass in “OPLElectionTest.csv”	expected	Borg was seated for the R party with 2 votes. Pike was seated for the D party with 3 votes. Foster was seated for the D party with 2 votes.	pass	
tc_014	An IRElection where two candidates are	BallotRedistributeTester.c	.csv file is available to	Run program, pass in	expected	50% of the time, candidate 1 is winning,	pass	

	tied at 50% botes each after two candidates are removed	sv	the program	“BallotRedistributeTEster.csv” when prompted		50% of the time candidate 2 is winning		
tc_015	Test FileProcessor’s ability to return an empty list if no ballots are in the election file	emptyelection.csv	The election file does not contain any ballots	Create FileProcessor object, give FileProcessor a file pointer and empty list of candidates.	expected	File is opened and FIleProcessor gives an output of an empty list	pass	
tc_016	Test giveBallot’s ability to fail to give a ballot to a candidate that is not in the running	n/a	IRCandidate has been created and then removed from the election using the removeCandidate() method	Run giveBallot() on the candidate with an arbitrary IRBallot object passed in. Then call getBallotCount() on the candidate	expected	Ballot is not given to the candidate and getBallotCount() returns 0.	pass	
tc_017	Test FileProcessor’s ability to return an empty list if no election type is specified at the top of the election file	nooplorir.csv	The election file does not contain any ballots	Run program, pass in “nooplorir.csv”, create FileProcessor object, run processFile()	expected	File is opened, fileprocessor is created, empty list is returned	pass	
tc_018	Test the OPL system so that candidate ballots are accurately counted and eliminated in each	OPLtest2.csv	The election file contains ballots and candidates are out of	Read file, ballots are tallied for candidates, candidates are reordered	expected	0Rob was seated for the R party with 3 votes. John was seated for the D party with 4 votes.	pass	

	round		order to be reordered					
tc_019	Test giveBallot()’s ability to give a ballot to the specified candidate	Name: “Ron Johnson” Party: “G” Ballot Index: “1” Name: “Jon Rohnson” Party: “R” Ballot Index: “4” Name: “Candice Tyler” Party: “D” Ballot Index: “3” Name: “Audrey Owens” Party: “I” Ballot Index: “2”	Ron Johnson’s ballot count is 0	Create an arrayList of IRCandidate objects, create an IRBallot, give ballot to first candidate	expected	IRCandidate arrayList is created and instantiated with the IRCandidates as described in the test data, IRBallot object is created, Ron Johnson’s ballot ArrayList contains the ballot object	pass	
tc_020	Test removeCandidate()’s ability to redistribute ballots to the next candidate in the ballot.	Name: “Ron Johnson” Party: “G” Ballot Index: “1” Name: “Jon Rohnson” Party: “R” Ballot Index: “4” Name: “Candice Tyler” Party: “D” Ballot Index: “3” Name: “Audrey Owens” Party: “I” Ballot Index: “2”	None	Create an ArrayList of IRCandidate objects. Create an IRBallot Give ballot to first candidate Remove “Ron Johnson” from the race.	Expected	“Ron Johnson”’s ballot ArrayList contains no ballots and is empty. Ron Johnson’s total number of ballots is equal to 0. Ron Johnson’s ‘inRunning’ value is equal to false. “Jon Rohnson”’s ArrayList contains the ballot previously held by Ron Johnson. “Jon Rohnson”’s total number of ballots is equal to 1.	Pass	
tc_021	Test MPO election with 6 candidates and two seats, with a tie for the second seat	testing/MPOtest1.csv	None	Run program and input “file/path/to/MPOtest1.csv ”	Expected	Pike wins a seat with 3 votes, and either Borg or Foster win the second seat, both with 2 votes	Pass	

tc_022	Test MPO election with 6 candidates and 3 seats, tie for second place	testing/MPOtest2.csv	None	Run program and input "file/path/to/MPOtest2.csv"	Expected	Pike wins first seat with 3 votes, then either Borg or Foster win the second seat with 2 votes, and the other wins the 3rd seat, also with 2 votes	Pass	
tc_023	Test MPO election with 6 candidates and one seat, and a tie for first place	testing/MPOtest3.csv	None	Run program and input "file/path/to/MPOtest3.csv"	Expected	Either Pike or Smith win the seat	Pass	
tc_024	Test tie breaking with 2 candidates in MPO for fairness	MPOVotingSystemTest.java	JUnit is prepared	Run MPOVotingSystemTest.java JUnit tests	Expected	Each candidate in the tie is chosen with less than 5% deviation from 50% with 1000 repetitions	Pass	
tc_025	Test tie breaking with 3 candidates in MPO for fairness	MPOVotingSystemTest.java	JUnit is prepared	Run MPOVotingSystemTest.java JUnit tests	Expected	Each candidate in the tie is chosen with less than a 5% deviation from 33% with 1000 repetitions.	Pass	
tc_026	Test HeaderProcessor's ability to parse MPO header	MPOParsingTest.java	JUnit is prepared	Run MPOParsingTest.java JUnit test	Expected	MPOVotingSystem is created with candidates Pike (D) and Deutsch ®, expecting 6 ballots and 1 candidate to be seated.	Pass	
tc_027	Test FileProcessor's ability to parse MPO Ballots	MPOParsingTest.java	JUnit is prepared	Run MPOParsingTest.java JUnit test	Expected	Pike (D) has 2 votes and Deutsch (R) has 0	Pass	