Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19
Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: CPLTest_1
Test Description:

Check if CPL object is constructed correctly and winners are correctly determined.

CPLTest.java
cplTest1()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Assert that the number of parties has been correctly read from the file and assigned to a variable.	cpl_ballot1.csv cpl.numOfParties	True	True	4 in this case.
2	Assert that the number of ballots has been correctly read from the file and assigned to a variable.	cpl_ballot1.csv cpl.numOfBallots	True	True	13 in this case.
3	Assert that the number of seats has been correctly read from the file and assigned to a variable.	cpl_ballot1.csv cpl.numOfSeats	True	True	7 in this case.
4	Assert that the number of candidates has been correctly read from the file and assigned to a variable.	cpl_ballot1.csv cpl.numOfCandidates	True	True	16 in this case.
5	Assert that the winners of the election have been correctly selected by the RunVote() method.	expectedWinners cpl.winners	True	True	

Post condition(s) for Test:

CPL object has been created and has decided the winners for the election given the CPL algorithm.

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit X__ System __

Test Date: 11/18/19

Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: CPLTest_2

Test Description:

Check if CPL object is constructed correctly and winners are correctly determined when ~55% of the votes are for the Democratic party.

CPLTest.java
cplTest2()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Assert that the number of parties has been correctly read from the file and assigned to a variable.	cpl_majorityDemocrat.csv cpl.numOfParties	True	True	4 in this case.
2	Assert that the number of ballots has been correctly read from the file and assigned to a variable.	cpl_majorityDemocrat.csv cpl.numOfBallots	True	True	23 in this case.
3	Assert that the number of seats has been correctly read from the file and assigned to a variable.	cpl_majorityDemocrat.csv cpl.numOfSeats	True	True	7 in this case.
4	Assert that the number of candidates has been correctly read from the file and assigned to a variable.	cpl_majorityDemocrat.csv cpl.numOfCandidates	True	True	16 in this case.
5	Assert that the winners of the election have been correctly selected by the RunVote() method.	expectedWinners cpl.winners	True	True	4 out of 5 democrats won out of the 7 seats.

Post condition(s) for Test:

CPL object has been created and has decided the winners for the election given the CPL algorithm.

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit X__ System __

Test Date: 11/18/19

Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: CPLTest_3

Test Description:

Check if CPL object is constructed correctly and winners are correctly determined when 100% of the votes are for the Democratic party.

CPLTest.java
cplTest3()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Assert that the number of parties has been correctly read from the file and assigned to a variable.	cpl_overwhelmingDemocrat.csv cpl.numOfParties	True	True	4 in this case.
2	Assert that the number of ballots has been correctly read from the file and assigned to a variable.	cpl_overwhelmingDemocrat.csv cpl.numOfBallots	True	True	20 in this case.
3	Assert that the number of seats has been correctly read from the file and assigned to a variable.	cpl_overwhelmingDemocrat.csv cpl.numOfSeats	True	True	5 in this case.
4	Assert that the number of candidates has been correctly read from the file and assigned to a variable.	cpl_overwhelmingDemocrat.csv cpl.numOfCandidates	True	True	16 in this case.
5	Assert that the winners of the election have been correctly selected by the RunVote() method.	expectedWinners cpl.winners	True	True	5 out of 5 democrats won out of the 7 seats.

Post condition(s) for Test:

CPL object has been created and has decided the winners for the election given the CPL algorithm.

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit X__ System __

Test Date: 11/18/19

Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: CPLTest_4

Test Description:
Check if RunVote() is randomly choosing a winner when there is a tie between 2 candidates for a CPL object.

CPLTest.java
cplTest4()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Constuct a CPL object and run RunVote() 1000 times for one file, and keep track of the number of winners between two candidates.	cpl_ties.csv	candidateOneWins and candidateTwoWins are both close to 500.	candidateOneWins and candidateTwoWins are both close to 500.	
2	Assert that the ratio candidateTwo wins the election is above 0.45.	candidateOneWins candidateTwoWins	True	True	
3	Assert that the ratio candidateOne wins the election is below 0.55.	candidateOneWins candidateTwoWins	True	True	
4					
5					

Post condition(s) for Test:

A CPL object will choose a particular candidate about 50% of the time when there is a tie between two candidates.

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _x_ System __

Test Date: 11/18/19

Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: CPLTest_5

Test Description:

Check if RunVote() for CPL runs a ballot with over 100,000 votes in under 5 minutes

CPLTest.java
cplTest5()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Constuct a CPL object with the ballot file that has over 100,000 ballots and call RunVote()	cpl_ballot_stress.csv	Cpl object creation and runVote() takes less than five menets	The runTime is less than 300,000 ms (five minutes.	
2	Assert that (start-end) is less than 300,000 ms	Start end	True	True	
3					
4					
5					

Post condition(s) for Test

The CPL object will be created and runVote() will be finished within five minutes.

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _x_ System __

Test Date: 11/18/19

Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: CPLTest_6

Test Description:

Check if RunVote() for CPL run a ballot with over 100,000 votes in under 5 minutes

CPLTest.java
cplTest6()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Constuct a CPL object with the ballot file that has over 100,000 ballots and call RunVote()	cpl_ballot_stress.csv	Cpl object creation and runVote()	Cpl Object created and runVote() called	
2	Assert that the winners are correct in the stress ballot	Cpl.winners	True	True	
3					
4					
5					

Post condition(s) for Test:

The CPL object will be created and runVote() will produce the correct winners for the ballot with over 100,000 votes

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19

Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: CPLTest_7

Test Description:
Check that CPL runVote algorithm is consistent when run iteratively

CPLTest.java
cplTest7()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Constuct a CPL object and run RunVote() 1000 times for one file, and keep track number of times a candidate wins	cpl_just_one_person.csv	CPL object created and runVote() called 1000 times, if expected candidate wins increment candidateOneWins	CPL object created and runVote() called 1000 times and candidateOneWins incremented 1000 times	
2	Assert that candidateOne won every single time	candidateOneWins	True	True	
3					
4					
5					

Post condition(s) for Test:

CPL object created and runVote() method called 1000 times to ensure consistency.

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19

Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: CPLTest_8

Test Description:
Check if RunVote() is randomly choosing a winner when there is a tie between 2 candidates for a CPL object.

CPLTest.java
cplTest8()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Constuct a CPL object and run RunVote() 1000 times for one file,	cpl_ballot1csv	CPL object created and runVote() called 1000 times,	CPL object created and runVote() called 1000 times	
2	Assert that the winners are correct for each step in the loop	Cpl.winners	True	True	Done 1000 times, for each step. Each runVote() call should yeild the same results
3					
4					
5					

Post condition(s) for Test:
CPL object created and runVote() method called 1000 times and the winners tested each time.

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19

Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: CPLTest_9

Test Description:
Check if RunVote() is randomly choosing a winner when there is a tie between 2 candidates for a CPL object.

Automated: yes__X__ no ___

CPLTest.java
cplTest9()

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Constuct a CPL object and run RunVote() 1000 times for one file,	cpl_majorityDemocrat.csv	CPL object created and runVote() called 1000 times,	CPL object created and runVote() called 1000 times	
2	Assert the number of parties is correct	cpl.numOfParties	True	True	Done 1000 times, for each step. Each runVote() call should yield the same results. Same for steps 2-6
3	Assert the number of ballots is correct	cpl.numOfBallots	True	True	
4	Assert the number of Seats is correct	cpl.numOfSeats	True	True	
5	Assert the number of candidates is correct	cpl.numOfCandidates	True	True	
6	Assert that the winners are correct for each step in the loop	Cpl.winners	True	True	

Post condition(s) for Test:
CPL object created and runVote() method called 1000 times with object attributes and the winners tested each time.

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19

Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: GetCorrectTerminalInput

Test Description:

Input a cpl_ballot file into the console to test GetTerminalInput()'s reading from terminal

DriverTerminal.java

GetCorrectTerminalInput()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Reroute System.in to be the tested filename	cpl_ballot1.csv	System.in will now be the filename	System.in will now be the filename	
2	Create a Driver object and test getTerminalInput()	Scanner object	Scanner reads the filename	Scanner reads the file name	The Scanner object exists within the GetTerminalOutput()
3	Assert input String within the Driver object is the filename	d.input	True	True	
4					
5					

Post condition(s) for Test:

GetTermnalOutput within the Driver class works as intended for a CPL ballot, and can store the filename

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19

Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: GetCorrectTerminalInput2

Test Description:

Input a opl ballot file into the console to test GetTerminalInput()'s reading from terminal

DriverTerminal2.java

GetCorrectTerminalInput2()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Reroute System.in to be the tested filename	opl_ballot1.csv	System.in will now be the filename	System.in will now be the filename	
2	Create a Driver object and test getTerminalInput()	Scanner object	Scanner reads the filename	Scanner reads the file name	The Scanner object exists within the GetTerminalOutput()
3	Assert input String within the Driver object is the filename	d.input	True	True	
4					
5					

Post condition(s) for Test:

GetTermnalOutput within the Driver class works as intended for an OPL ballot, and can store the filename

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit X __ System __

Test Date: 11/18/19
Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: OPLTest_1
Test Description:

Check if OPL object is constructed correctly and winners are correctly determined.

OPLTest.java
oplTest1()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Assert that the number of ballots has been correctly read from the file and assigned to a variable.	opl_ballot1.csv opl.numOfBallots	True	True	9 in this case.
2	Assert that the number of seats has been correctly read from the file and assigned to a variable.	opl_ballot1.csv cpl.numOfSeats	True	True	3 in this case.
3	Assert that the number of candidates has been correctly read from the file and assigned to a variable.	opl_ballot1.csv opl.numOfCandidates	True	True	6 in this case.
4	Assert that the winners of the election have been correctly selected by the RunVote() method.	expectedWinners opl.winners	True	True	

Post condition(s) for Test: OPL object has been created and has decided the winners for the election given the OPL algorithm.

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19
Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: OPLTest_2
Test Description:

Check if OPL object is constructed correctly and winners are correctly determined.

OPLTest.java
oplTest2()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Assert that the number of ballots has been correctly read from the file and assigned to a variable.	opl_majorityDemocrat.csv cpl.numOfBallots	True	True	29 in this case.
2	Assert that the number of seats has been correctly read from the file and assigned to a variable.	opl_majorityDemocrat.csv opl.numOfSeats	True	True	3 in this case.
3	Assert that the number of candidates has been correctly read from the file and assigned to a variable.	opl_majorityDemocrat.csv opl.numOfCandidates	True	True	6 in this case.
4	Assert that the winners of the election have been correctly selected by the RunVote() method.	opl_majorityDemocrat.csv opl.winners	True	True	

Post condition(s) for Test: OPL object has been created and has decided the winners for the election given the OPL algorithm.

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19
Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: OPLTest_3
Test Description:

An OPL ballot with a tie is tested 1000 times to ensure that the candidate that wins is random

OPLTest.java
oplTest3()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Constuct a OPL object and run RunVote() 1000 times for one file, and keep track of the number of winners between two candidates.	opl_two_candidates_ties.csv	candidateOneWins and candidateTwoWins are both close to 500.	candidateOneWins and candidateTwoWins are both close to 500.	Constuct a OPL object and run RunVote() 1000 times for one file, and keep track of the number of winners between two candidates.
2	Assert that the ratio candidateTwo wins the election is above 0.45.	candidateOneWins candidateTwoWins	True	True	Assert that the ratio candidateTwo wins the election is above 0.45.
3	Assert that the ratio candidateOne wins the election is below 0.55.	candidateOneWins candidateTwoWins	True	True	Assert that the ratio candidateOne wins the election is below 0.55.
4					

Post condition(s) for Test: A OPL object will choose a particular candidate about 50% of the time when there is a tie between two candidates.

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19
Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: OPLTest_4
Test Description:

In this OPL ballot, one candidate gets all the votes but more than one seat is allocated. The second winners should be a member of the same party

OPLTest.java
oplTest4()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Assert that the candidate that got all the votes is the first winner	opl_only_one_vote.csv candidateOne	True	True	
2	Assert that the second Democrat that got no votes is the second winner	opl_only_one_vote.csv CandidateTwo	True	True	This candidate did not get any ballots but still wins
3					
4					

Post condition(s) for Test: OPL object has been created and the correct winners have been decided by RunVote()

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19
Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: OPLTest_5
Test Description:

Check if OPL RunVote() runs a ballot with over 100,000 votes in under 5 minutes

OPLTest.java
oplTest5()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Construct an OPL object with the ballot file that has over 100,000 ballots cal call RunVote()	opl_ballot_stress.csv	OPL object creation and runVote() takes less than five minutes	The runTime is less than 300,000 ms (five minutes)	
2					
3					
4					

Post condition(s) for Test: The CPL object will be created and runVote will be finished within five minutes

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19
Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: OPLTest_6
Test Description:

Check if OPL RunVote() runs a ballot with over 100,000 votes and the winners are correct

OPLTest.java
oplTest6()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Construct an OPL object with the ballot file that has over 100,000 ballots cal call RunVote()	opl_ballot_stress.csv	OPL object creation and runVote() called	OPL object creation and runVote() called	
2	Tested that winners are correct in the stress ballot	Opl.winners	True	True	
3					
4					

Post condition(s) for Test: The OPL object will be created and runVote will produce the correct winners

Project Name: Project 1: Voting System **Team#28**

Test Stage: Unit _X_ System __

Test Date: 11/18/19

Name(s) of Testers: Ankith Bhat, Gabriel Lee, Philip Neff, Neil Patel

Test Case ID#: OPLTest_7

Test Description:

Check if OPL RunVote() run a ballot where only one person is contesting

OPLTest.java
oplTest7()

Automated: yes__X__ no ___

Results: Pass __X__ Fail_____

Preconditions for Test: GetTerminalInput() has run and csv file exists in directory.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Get the ballot file	opl_just_one_person.csv	OPL object creation and runVote() called	OPL object creation and runVote() called	
2	Test that only one candidate won, and the correct candidate won	Opl.winners	True	True	
3					
4					

Post condition(s) for Test: The OPL object will be created and runVote will produce the correct winner

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit __ System _X_

Test Date: 11/18/2019
Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_1
Test Description:

Check that an audit file is generated and correct for an OPL vote

Automated: yes___ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test: Appropriate CSV ballot file exists in directory

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Enter OPL_ballot_1.csv into terminal input	OPL_ballot_1.csv	System will display results to screen	System displayed correct results	
3	Check audit file has been generated with unique file name	Audit	Audit generated in directory with unique file name	Audit generated in directory with unique file name	
4	Open audit file and check that information is correct	Audit file	Results displayed to screen and candidate information on winners	Results displayed to screen and candidate information on winners	

Post condition(s) for Test: Audit file exists and contains correct vote information

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit __ System _X_

Test Date: 11/18/2019
Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_2
Test Description:

Check that a media report is generated and correct for an OPL vote

Automated: yes___ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test: Appropriate CSV ballot file exists in directory

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Enter OPL_ballot_1.csv into terminal input	OPL_ballot_1.csv	System will display results to screen	System displayed correct results	
3	Check audit file has been generated with unique file name	Audit	Audit generated in directory with unique file name	Audit generated in directory with unique file name	
4	Open audit file and check that information is correct	Audit file	Results displayed to screen and candidate information on winners	Results displayed to screen and candidate information on winners	
5	Enter 'Y' into terminal to generate media report	Y	Media Report generated in directory with unique file name	Media Report generated in directory with unique file name	
6	Open media report and check that information is correct	Media Report	Results displayed to screen	Results displayed to Screen	

Post condition(s) for Test: Media Report exists and contains correct vote information

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit __ System _X_

Test Date: 11/18/2019
Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_3
Test Description:

Check that an audit file is generated and correct for a CPL vote

Automated: yes___ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test: Appropriate CSV ballot file exists in directory

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Enter CPL_ballot_1.csv into terminal input	CPL_ballot_1.csv	System will display results to screen	System displayed correct results	
3	Check audit file has been generated with unique file name	Audit	Audit generated in directory with unique file name	Audit generated in directory with unique file name	
4	Open audit file and check that information is correct	Audit file	Results displayed to screen and candidate information on winners	Results displayed to screen and candidate information on winners	

Post condition(s) for Test: Audit file exists and contains correct vote information

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit __ System _X_

Test Date: 11/18/2019

Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_4

Test Description:

Check that a media report is generated and correct for an OPL vote

Automated: yes___ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test: Appropriate CSV ballot file exists in directory

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Enter CPL_ballot_1.csv into terminal input	CPL_ballot_1.csv	System will display results to screen	System displayed correct results	
3	Check audit file has been generated with unique file name	Audit	Audit generated in directory with unique file name	Audit generated in directory with unique file name	
4	Open audit file and check that information is correct	Audit file	Results displayed to screen and candidate information on winners	Results displayed to screen and candidate information on winners	
5	Enter 'Y' into terminal to generate media report	Y	Media Report generated in directory with unique file name	Media Report generated in directory with unique file name	
6	Open media report and check that information is correct	Media Report	Results displayed to screen	Results displayed to Screen	

Post condition(s) for Test: Media Report exists and contains correct vote information

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit X_ System __

Test Date: 11/18/2019

Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_5

Test Description:

Check if incorrect input results in a new prompt for file input with appropriate error message.

Automated: yes__X_ no ___

Results: Pass __X___ Fail_____

Preconditions for Test: GetTerminalInput has displayed prompt to screen

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Press Enter when prompted for file name	None	"Please enter a valid file name (too long or too short)"	Please enter a valid file name (too long or too short)	
2					
3					
4					

Post condition(s) for Test:

System is waiting for keyboard input.

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit X_ System __

Test Date: 11/18/2019

Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_6

Test Description:

Check if incorrect input results in a new prompt for file input with appropriate error message.

Automated: yes__X_ no ___

Results: Pass __X___ Fail_____

Preconditions for Test: GetTerminalInput has displayed prompt to screen

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Press Space 6 times when prompted for file name	None	Please enter a valid file name (doesn't exist)	Please enter a valid file name (doesn't exist)	
2					
3					
4					

Post condition(s) for Test:

System is waiting for keyboard input.

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit X_ System __

Test Date: 11/18/2019

Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_7

Test Description:

Check if incorrect input results in a new prompt for file input with appropriate error message.

Automated: yes__X_ no ___

Results: Pass __X___ Fail_____

Preconditions for Test: GetTerminalInput has displayed prompt to screen

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Input 256 characters when prompted for file name	None	Please enter a valid file name (too long or too short)	Please enter a valid file name (too long or too short)	
2					
3					
4					

Post condition(s) for Test:

System is waiting for keyboard input.

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit X_ System __

Test Date: 11/18/2019

Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_8

Test Description:

Check if incorrect input results in a new prompt for file input with appropriate error message.

Automated: yes__X_ no ___

Results: Pass __X___ Fail_____

Preconditions for Test: GetTerminalInput has displayed prompt to screen

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Enter 'not_a_csv.txt'	not_a_csv.txt	Please enter a valid file name (not a csv)	Please enter a valid file name (not a csv)	
2					
3					
4					

Post condition(s) for Test:

System is waiting for keyboard input.

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit __ System _X_

Test Date: 11/18/2019

Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_9

Test Description:

Check that a media report is not generated for an OPL vote

Automated: yes___ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test: Appropriate CSV ballot file exists in directory

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Enter OPL_ballot_1.csv into terminal input	OPL_ballot_1.csv	System will display results to screen	System displayed correct results	
3	Check audit file has been generated with unique file name	Audit	Audit generated in directory with unique file name	Audit generated in directory with unique file name	
4	Open audit file and check that information is correct	Audit file	Results displayed to screen and candidate information on winners	Results displayed to screen and candidate information on winners	
5	Enter 'N' into terminal to reject generating a media report	N	Directory remains unchanged	Directory remains unchanged.	

Post condition(s) for Test: Program stops and directory remains unchanged.

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit __ System _X_

Test Date: 11/18/2019

Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_10

Test Description:

Check that a media report is not generated for an CPL vote

Automated: yes___ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test: Appropriate CSV ballot file exists in directory

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Enter CPL_ballot_1.csv into terminal input	CPL_ballot_1.csv	System will display results to screen	System displayed correct results	
3	Check audit file has been generated with unique file name	Audit	Audit generated in directory with unique file name	Audit generated in directory with unique file name	
4	Open audit file and check that information is correct	Audit file	Results displayed to screen and candidate information on winners	Results displayed to screen and candidate information on winners	
5	Enter 'N' into terminal to reject generating a media report	N	Directory remains unchanged	Directory remains unchanged.	

Post condition(s) for Test: Program stops and directory remains unchanged.

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit __ System _X_

Test Date: 11/18/2019

Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_11

Test Description:

Check that in the case of a tie for an OPL vote, winner is picked at random

Automated: yes___ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test: Appropriate CSV ballot file exists in directory

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Enter OPL_all_tied.csv into prompt	OPL_all_tied.csv	A winner is chosen	A winner is chosen	
2	Repeat Step 1 19 more times	OPL_all_tied.csv	At least one of those rounds has a different winner than the initial winner	10 of the rounds had a different winner	While possible that all 20 rounds have the exact same winner, it is very unlikely.
3					
4					

Post condition(s) for Test:

Project Name: Project 1: Voting System **Team# 28**

Test Stage: Unit __ System _X_

Test Date: 11/18/2019

Name(s) of Testers: Ankith Bhat, Neil Patel, Philip Neff, Gabriel Lee

Test Case ID#: SYS_12

Test Description:

Check that in the case of a tie in a CPL vote, winner is picked at random

Automated: yes___ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test: Appropriate CSV ballot file exists in directory

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Enter CPL_all_tied.csv into prompt	CPL_all_tied.csv	A winner is chosen	A winner is chosen	
2	Repeat Step 1 19 more times	CPL_all_tied.csv	At least one of those rounds has a different winner than the initial winner	14 of the rounds had a different winner	While possible that all 20 rounds have the exact same winner, it is very unlikely.
3					
4					

Post condition(s) for Test: