Project Team# 5	Name	Project	1:	Voting	System		
Test Stage:	Unit	System <u>X</u>					
Test Case ID	)#: IR_001		Name(s)	of Testers: Trevor			
Test Descrip	otion:						
example file the winner to	IR test Description with a copy of the example file given on Canvas. Should print the winner to screen. Audit and media files should be created in /src directory.						
			(what file	where are you storing and the name of the unctions being used	e		
				ile is "IR.txt" and it is s rectory inside our repo			
Automated:	yes	no _ <u>X</u>					
Results: Pa	ss <u>X</u>	Fail					
Precondition	is for Test: Th	e "IR.txt" exists ar	nd is correctly	formatted.			

Ste			Expected	Actual	Notes
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Compile all files.	Open terminal. Use javac to compile /src files.	Compiles with no errors	Compiles with no errors	
١ .	Run the application	Open terminal. Navigate to the working directory. Use "java SunshineApp.jav a" to run the application	GUI should pop up on the user's screen with a window to input the file name.	GUI popped up on the user's screen with a window to input the file name.	
3	Enter the File containing election information	Where prompted by the GUI, enter "IR.txt" into the search bar.	The system should accept the file and run the election. A winner will be displayed to the GUI.	The election run was successful. Rosen won the election and his name and party were displayed on the GUI.	
4	Check for Media/Audit File	Open the /src directory and search for the audit and media files	They should both contain their relevant information.	Upon opening them, both files contained the needed information.	

The GUI displays text that tells some information about the election. It shows that the election was an IR election. It also shows the number of ballots cast and the number of candidates that ran during the election. Finally, the winner, their party, and their votes are displayed.

Audit and media files were created in the /src directory. The system has not changed.

Project Name: Team# 5	Project	1:	Voting	System
Test Stage: Unit System <u>)</u>	<u>C</u> Test	t Date: 5/2	2/2021	
Test Case ID#: IR_002	Nan	ne(s) of Te	esters: Trevor	
Test Description:				
IR test Description with a CSV file the results in a 4 way tie after the initial distribution. I will run this file multiple to ensure that a winner is chosen rar due to the tiebreakers.	times ndomly Indi (who	at file) and hod/funct	e are you storing I the name of the ions being used	e
			"IR4way.txt" and directory of our re	
Automated: yes no <u>X</u> _				
Results: Pass <u>X</u> Fail				
Preconditions for Test: The "IR4way	ytie.txt" exists a	nd is corre	ctly formatted.	

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Compile all files.	Open terminal. Use javac to compile /src files.	Compiles with no errors	Compiles with no errors	
2	Run the application	Use "java	GUI should pop up on the user's screen with a window to input the file name.	GUI popped up on the user's screen with a window to input the file name.	
3	Enter the File containing election information	Where prompted by the GUI, enter "IR4way.txt" into	The system should accept the file and run the election. A random winner will be displayed to the GUI. This occurs because of the four way tie.	The election was run multiple times and a different winner was chosen on different runs. This implies that the winner chosen is random which is good.	
4	Check for Media/Audit File	Open the /src directory and search for the audit and media files	They should both contain their relevant information.	Upon opening them, both files contained the needed information.	

The GUI displays text that tells some information about the election. It shows that the election was an IR election. It also shows the number of ballots cast and the number of candidates that ran during the election. Finally, the winner, their party, and their votes are displayed.

Audit and media files were created in the /src directory. The system has not changed.

Project Nar Team# 5	me: Proj	ect 1:	Voting	System
Test Stage: Unit	System <u>X</u>	Test Date:	5/2/2021	
Test Case ID#: IR_003		Name(s) of	Testers: Trevor	
Test Description:				
IR test Description with a results in a 2 way tie after distribution. A winner show random using the flipCoin	the vote uld be chosen at			
		(what file) a	ere are you storing nd the name of the ctions being used	e
			is "IR2waytie.txt" ar e /testing directory	
Automated: yes	no <u>X</u>			
Results: Pass X	Fail			
Preconditions for Test:	Γhe "IR2waytie.txt" e	xists and is cor	rectly formatted.	

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Compile all files.	Open terminal. Use javac to compile /src files.	Compiles with no errors	Compiles with no errors	
2	Run the application	Use "java	GUI should pop up on the user's screen with a window to input the file name.	GUI popped up on the user's screen with a window to input the file name.	
	Enter the File containing election information	Where prompted by the GUI, enter "IR2waytie.txt" into the search	The system should accept the file and run the election. A random winner will be displayed to the GUI. This occurs because of the four way tie.	The election was run multiple times and a different winner was chosen on different runs. This implies that the winner chosen is random which is good.	
4	Check for Media/Audit File	Open the /src directory and search for the audit and media files	They should both contain their relevant information.	Upon opening them, both files contained the needed information.	

The GUI displays text that tells some information about the election. It shows that the election was an IR election. It also shows the number of ballots cast and the number of candidates that ran during the election. Finally, the winner, their party, and their votes are displayed.

Audit and media files were created in the /src directory. The system has not changed.

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage: Unit	System <u>X</u>	Test	Date: 5/2	2/2021	
Test Case ID#: IR	_004	Nam	e(s) of Te	esters: Trevor	
Test Description:					
IR test Description of results in a ballot be ballot is owned by conditional alternate vote being Candidate 1 is elimicandidate 4 is elimical ballot in question to for no more candidate.	eing deleted. The o candidate 4 with the g placed for candid inated first. Then, nated. This causes be deleted since i	original e only ate 1. s the			
		(wha	t file) and	re are you storing d the name of the ions being used	9
		locat	ed in the a	"IReliminateballot /testing directory of e entire system is	of our
Automated: yes_	no <u>_X</u>				
Results: Pass	<u>(</u> Fail				
Preconditions for 1	<b>Test:</b> The "IRelimin	ateballot.txt" ex	kists and i	s correctly format	ted.

Cto			Expected	Actual	
Ste p	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Compile all files.	Open terminal. Use javac to compile /src files.	Compiles with no errors	Compiles with no errors	
2	Run the application	Open terminal. Navigate to the working directory. Use "java SunshineApp.jav a" to run the application	GUI should pop up on the user's screen with a window to input the file name.	GUI popped up on the user's screen with a window to input the file name.	
3	Enter the File containing election information	Where prompted by the GUI, enter "IReliminateballot .txt" into the search bar.	The GUI will display information regarding the winner of the election. Upon checking the audit file we can see that the ballot was lost after candidate 4 is eliminated.	The GUI displayed the relevant information and the audit file did indicate that the ballot was deleted. This was a success.	
4	Check for Media/Audit File	Open the /src directory and search for the media and audit files that have been generated.	They should both contain their relevant information.	Upon opening them, both files contained the needed information.	

The GUI displays text that tells some information about the election. It shows that the election was an IR election. It also shows the number of ballots cast and the number of candidates that ran during the election. Finally, the winner, their party, and their votes are displayed.

Audit and media files were created in the /src directory. The system has not changed

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage:	Unit	System <u>X</u>	Test Date: 5/2	2/2021	
Test Case ID	#: OPL_00	1	Name(s) of Te	esters: Trevor	
Test Descrip	tion:				
OPL election copied off the					
			(what file) and	e are you storing d the name of the ions being used.	•
				"OPL.txt" and it is ectory of our repo	
Automated:	yes	no <u>X</u>			
Results: Pa	ss <u>X</u>	Fail			
Precondition	s for Test: ┐	The "OPL.txt" exists ar	nd is correctly fo	ormatted.	

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Compile all files.	Open terminal. Use javac to compile /src files.	Compiles with no errors	Compiles with no errors	
2	Run the application	Open terminal. Navigate to the working directory. Use "java SunshineApp.jav a" to run the application	GUI should pop up on the user's screen with a window to input the file name.	GUI popped up on the user's screen with a window to input the file name.	
3	Enter the File containing election information	Where prompted by the GUI, enter "OPL.txt" into the search bar.	The system should accept the file input and run the election. The winners of seats should be shown to the screen.	The election was successfully run. All winners were displayed to the screen.	
4	Check for Media/Audit File	Open the /src directory and search for the audit and media files.	They should both contain their relevant information.	Upon opening them, both files contained the needed information including intermediate test descriptions.	

The GUI displays the names and parties of all candidates that were awarded a seat. The media and audit files were created. The information in the media file matches the information that is displayed to the user via the GUI. The audit file contains information regarding intermediate steps including vote assignments and seat awards.

Both the audit and media files are created and stored in the /src directory. The system has not changed.

Project Name: Team# 5	Project	1:	Voting	System
Test Stage: Unit System	n <u>X</u> Test	Date: 5/2	2/2021	
Test Case ID#: OPL_002	Nan	ne(s) of Te	sters: Trevor	
Test Description:				
OPL election test in which a party seats than it has candidates to ass to. It needs to reassign seats to a caparty.	sign them			
	(what	at file) and	e are you storing If the name of the ions being used	e
	and		"OPLTOOMANYS d in the /testing d	
Automated: yes no X	_			
Results: Pass X Fail				
Preconditions for Test: The "OPL	TOOMANYSEATS	S.txt" exists	and is correctly	formatted.

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Compile all files.	Open terminal. Use javac to compile /src files.	Compiles with no errors	Compiles with no errors	
2	Run the application	Open terminal. Navigate to the working directory. Use "java SunshineApp.jav a" to run the application	GUI should pop up on the user's screen with a window to input the file name.	GUI popped up on the user's screen with a window to input the file name.	
3	Enter the File containing election information	Where prompted by the GUI, enter "OPLTOOMANY SEATS.txt" into the search bar.	, ,	The election was successfully run. All winners were displayed to the screen.	
4	Check for Media/Audit File	Open the /src directory and search for the audit and media files.	They should both contain their relevant information. The media file should show that a seat was temporarily awarded and then redistributed.	Upon opening them, both files contained the needed information including intermediate test descriptions.	

The GUI displays the names and parties of all candidates that were awarded a seat. The media and audit files were created. The information in the media file matches the information that is displayed to the user via the GUI. The audit file contains information regarding intermediate steps including vote assignments and seat awards.

Audit and media files are created and stored in the /src directory.

Project Name: Team# 5	Project	1:	Voting	System	
Test Stage: Unit System <u>X</u>	Tes	t Date: 5/2	2/2021		
Test Case ID#: OPL_003	Nan	ne(s) of Te	esters: Trevor		
Test Description:					
OPL Election test in which a party is awarded a seat. However, the top 2 candidates in that party are tied so a tiebreaker must be used to determine who gets the seat. File will be passed in multiple times to ensure the tiebreaker is random.  Indicate where are you storing the tests					
	•	•	d the name of the ions being used		
	_		"OPLTIE.txt" and directory of our re		
Automated: yes no X_					
Results: Pass X Fail					
Preconditions for Test: The "OPLTIE	E.txt" exists and	l is correct	ly formatted.		

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Compile all files.	Open terminal. Use javac to compile /src files.	Compiles with no errors	Compiles with no errors	
2	Run the application	Open terminal. Navigate to the working directory. Use "java SunshineApp.jav a" to run the application	GUI should pop up on the user's screen with a window to input the file name.	GUI popped up on the user's screen with a window to input the file name.	
3	Enter the File containing election information	Where prompted by the GUI, enter "OPLTIE.txt" into the search bar.	The system should accept the file input and run the election. The winners of seats should be shown to the screen.	The election was successfully run. All winners were displayed to the screen. Multiple runs resulted in differing winners which is good.	
4	Check for Media/Audit File	Open the /src directory and search for the audit and media files.	They should both contain their relevant information. The audit file should indicate that a tie has occurred.	Upon opening them, both files contained the needed information including intermediate test descriptions. Audit file showed that a tie occurred.	

The GUI displays the names and parties of all candidates that were awarded a seat. The media and audit files were created. The information in the media file matches the information that is displayed to the user via the GUI. The audit file contains information regarding intermediate steps including vote assignments and seat awards.

Audit and Media files were created and placed in the /src directory.

Project Name: Team# 5	Project	1:	Voting	System
Test Stage: Unit System X	Test	Date: 5/2	2/2021	
Test Case ID#: OPL_004	Nam	ne(s) of Te	sters: Trevor	
Test Description:				
OPL Election in which there are more s available than candidates running.	seats			
	(wha	at file) and	e are you storing I the name of the ions being used	9
	and		"OPLTOOMANYS d in the /testing d	
Automated: yes no X_				
Results: Pass X Fail				
Preconditions for Test: The "OPLTOO	MANYSEATS	62.txt" exis	ts and is correctly	formatted.

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Compile all files.	Open terminal. Use javac to compile /src files.	Compiles with no errors	Compiles with no errors	
2	Run the application	Open terminal. Navigate to the working directory. Use "java SunshineApp.jav a" to run the application	GUI should pop up on the user's screen with a window to input the file name.	GUI popped up on the user's screen with a window to input the file name.	
3	Enter the File containing election information	Where prompted by the GUI, enter "OPLTOOMANY SEATS2.txt" into the search bar.	The system should accept the file input and run the election. The winners of seats should be shown to the screen. There will be an extra seat	The election was successfully run. All winners were displayed to the screen.	
4	Check for Media/Audit File	Open the /src directory and search for the audit and media files.	They should both contain their relevant information. The media file should show that a seat was temporarily awarded and then redistributed.	Upon opening them, both files contained the needed information including intermediate test descriptions.	

The GUI displays the names and parties of all candidates that were awarded a seat. The media and audit files were created. The information in the media file matches the information that is displayed to the user via the GUI. The audit file contains information regarding intermediate steps including vote assignments and seat awards.

Audit and media files are created and stored in the /src directory

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage: Unit	t _X_ System	. Test	Date: 5/2	2/2021	
Test Case ID#:	SUNSHINE_TEST_	_001 Nam	e(s) of Te	sters: Tom	
Test Description	:				
Unit test for the Si constructor is test	unshineApp Class. T ed.	The			
		(wha	at file) and	e are you storing I the name of the ions being used	9
		it is I		"TestSunshineAp the /src directory	
Automated: yes	_X_ no				
Results: Pass _	X Fail				
	r <b>Test:</b> The environm " and "TestSunshine <i>l</i>	•		ly for JUnit testiກເ	ე. The

Ste p #	Test Step Description	Test Data		Actual Result	Notes
1		Open terminal. Use javac to compile /src files.	Compiles with no errors	Compiles with no errors	
2		Use the Junit framework to run the test.	Junit should respond with a green "Passed" message	Junit responded that the unit test was passed.	

The Junit GUI has displayed a "Passed" message. The system has not changed.

Project Team# 5	Name:	Project	1:	Voting	System		
Test Stage: Unit	_X_ System _	_ Test	Date: 5/2	2/2021			
Test Case ID#: C	ANDIDATE_TEST_	_001 Nan	ne(s) of Te	esters: Tom			
Test Description:							
Unit test for the Candidate Class. The constructor is tested. Also tested are getName(), getParty(), getBallots(), getNumBallots(), and getParty() are tested.							
		(wh	at file) and	e are you storing I the name of the ions being used	9		
				"TestCandidate.ja src directory of o			
Automated: yes_	<u>X</u> no						
Results: Pass	X Fail						
Preconditions for "Candidate.java" an		•	ed correct	ly for JUnit testiກເ	g. The		

Ste p #	Test Step Description	Test Data		Actual Result	Notes
1	Compile all files.	- P - · ·	Compiles with no errors	Compiles with no errors	
2	Run the Unit Test	Use the Junit framework to run the unit test.	Junit window should show a "Passed" message	Junit showed a "Passed" message	

The Junit GUI has displayed a "Passed" message. The system has not changed.

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage: Unit	t _X_ System _	Tes	t Date: 5/2	2/2021	
Test Case ID#: C	ANDIDATE_TEST	Γ_002 <b>N</b> an	ne(s) of Te	esters: Tom	
Test Description:	:				
	andidate Class. Th is tested. Also tes				
		(wh	at file) and	re are you storing d the name of the tions being used	e
				"TestCandidate.ja /src directory of o	
Automated: yes	<u>X</u> no				
Results: Pass _	X Fail				
Preconditions for "Candidate.java" ar		_	ired correc	ctly for JUnit testin	ng. The

Ste p #	Test Step Description	Test Data		Actual Result	Notes
1	Compile all files.	- P - · ·	Compiles with no errors	Compiles with no errors	
2	Run the Unit Test	Use the Junit framework to run the unit test.	Junit window should show a "Passed" message	Junit showed a "Passed" message	

The Junit framework displayed a pass message. The system has not changed.

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage: Unit	t _X_ Systen	n Test	Date: 5/	2/2021	
Test Case ID#: C	CANDIDATE_TE	ST_003 Nan	ne(s) of Te	esters: Tom	
Test Description	:				
Unit test for the CadeleteBallot functingetNumBallots(), and getBallots().	on is tested. Also				
		(wh	at file) an	re are you storing d the name of the tions being used	9
				"TestCandidate.ja /src directory of o	
Automated: yes	<u>X</u> no				
Results: Pass _	X Fail_				

Sto			Expected	Actual	
Ste p	Test Step	Test	Result	Result	Notes
#	Description	Data			
1		1000	Compiles with no errors	Compiles with no errors	
2	Run the Unit Test	Use the Junit	l	Junit showed a "Passed" message	

The Junit framework indicated that the test was passed. The system has not changed.

Project Name: Team# 5	Project	1:	Voting	System			
Test Stage: Unit _X_ System	Test	Date: 5/	2/2021				
Test Case ID#: BALLOT_TEST_001 Name(s) of Testers: Tom							
Test Description:							
Unit test for the Ballot Class. The Ballot constructor is tested. Also tested is getChoices().	ot						
	(wha	Indicate where are you storing the tests (what file) and the name of the method/functions being used.					
			"TestBallot.java" a /src directory of o				
Automated: yes_X_ no							
Results: Pass X Fail							
Preconditions for Test: The environment is configured correctly for JUnit testing. The "Ballot.java" and "TestBallot.java" exist.							

Ste p #	Test Step Description	Test Data		Actual Result	Notes
1	Compile all files.	- P - · ·	Compiles with no errors	Compiles with no errors	
2	Run the Unit Test	Use the Junit framework to run the unit test.	Junit window should show a "Passed" message	Junit showed a "Passed" message	

The Junit framework indicated that the test was passed. The system has not changed.

Project Team# 5	Name:	Project	1:	Voting	System			
Test Stage: Unit	_X_ System _	_ Test	Date: 5/2	2/2021				
Test Case ID#: BA	ALLOT_TEST_002	2 Nam	e(s) of Te	sters: Tom				
Test Description:								
	Unit test for the Ballot Class. The getChoices() function is tested.							
		(wha	t file) and	e are you storing I the name of the ions being used	e			
				"TestBallot.java" a src directory of o				
Automated: yes_	<u>X</u> no							
Results: Pass	<u>X</u> Fail							
<b>Preconditions for Test:</b> The environment is configured correctly for JUnit testing. The "Ballot.java" and "TestBallot.java" exist.								

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1			Compiles with no errors	Compiles with no errors	
2		Use the Junit	Junit window should show a "Passed" message	Junit showed a "Passed" message	

The Junit framework indicated that the test was passed. The system has not changed.

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage: Unit	t <u>X</u> System _	Test	Date: 5/2	2/2021	
Test Case ID#: OPL_TEST_PRO	CESS_FILE_001	Nam	ne(s) of Te	esters: Cole	
Test Description	:				
for the opl election gets the number of	nat it properly create	perly			
		(wha	at file) and	e are you storing I the name of the ions being used	9
		Test	OplElectio	ssFile is stored in n class in src, and OplElection.	
Automated: yes	X no				
Results: Pass _	X Fail				
	r <b>Test</b> : JUnit proper ndidate.java exist. T	•	-	•	-

Ste			•	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Create OplElection object		myElection is not null	myElection is not null	
2	Call processFile() method		f	processFile runs with no errors	
	Check that the file was processed correctly		correctly assigns the number of	processFile correctly assigns the number of balltos, seats, candidates, etc.	

The correct number of seats, ballots, candidates are obtained, as well as creating the proper Candidate objects and adding them to their correct party.

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage: Unit	<u>X</u> System	Tes	: Date: 5/	2/2021	
Test Case ID#: OPL_TEST_CON	STRUCTOR_001	Nan	ne(s) of Te	esters: Cole	
Test Description:					
Tests that the cons an OplElection ob variables properly.	ject and assigns t				
		(wh	at file) and	re are you storing d the name of the ions being used	e
		the	TestOplEle	nBallots method is ection class in src, OplElection.	
Automated: yes	<u>X</u> no				
Results: Pass	X Fail				
Preconditions for Candidate.java, and TestIrElection.java.			•	•	•

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	Create OpIElection Object		myElection is not null	myElection is not null	
	Check that file assigns class variables		read and assigns	The file is properly read and assigns the right values	

Creates the candidates properly and is able to correctly assign ballots to those candidates.

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage: Unit	X System	Test	Date: 5/2	2/2021	
Test Case ID#: OPL_TEST_ASSI	GN_BALLOTS_001	Nam	e(s) of Te	esters: Cole	
Test Description:					
		(wha	t file) and	e are you storing the name of the ions being used	9
		Test	OplElectio	ructor method is s in class in src, and ethod from OplEle	d uses the
Automated: yes_	_X_ no				
Results: Pass	<u>X</u> Fail				
	Test: JUnit compile ist. The "OPL.txt" file		-		Election.java.

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Create OplElection object		myElection is not null	myElection is not null	
1 ~	Call processFile() method		The file is properly processed	The file is properly processed	
	Check that candidates were created and ballots assigned			The candidates were properly created and assigned the proper ballots	

The proper candidates were created and assigned to the right party, with the correct ballots assigned to each candidate.

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage: Unit	X System _	_ Test	Date: 5/2	2/2021	
Test Case ID#: OPL_TEST_RUN	_ELECTION_001	Nam	e(s) of Te	esters: Cole	
Test Description:					
Tests that the runE OplElection runs p the election.	Election method in properly and correct	ly runs			
		(wha	t file) and	re are you storing d the name of the ions being used	9
		Test	OplElectio	ection method is an class in src, and OplElection.	
Automated: yes_	<u>X</u> no				
Results: Pass	<u>X</u> Fail				
Preconditions for Party.java, and Can TestOplElection.jav			•	•	-

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Create OplElection object		myElection is not null	myElection is not null	
2	Call runElection() method		"	runElection() runs and doesn't throw any errors	
3	Check that correct winners are declared		The correct winners are declared	The correct winners are declared	

The correct winners were declared from the election file.

ı	Project Na Team# 5	me: Pro	ject 1:	Voting	System
	Test Stage: Unit X	System	Test Date: 5/2/	2021	
	Test Case ID#: OPL_TEST_RUN_ELE(	CTION_TIE_001	Name(s) of Tes	sters: Cole	
	Test Description:				
	Runs an OPL election we tile within a party that was coin flip is used to determine the candidate is awarded the	s awarded a seat. A mine which	Indicate where	are you storin	n the tests
			(what file) and	are you storing the name of the ons being used	<b>e</b>
-1				otionTio mothod	is stored in
			The testRunEle the TestOplElec methods from C	tion class in src,	
	Automated: yes_X_	no	the TestOplElec	tion class in src,	
0	Automated: yes <u>X</u> Results: Pass <u>X</u>		the TestOplElec	tion class in src,	
			the TestOplElec	tion class in src,	

Ste			Expected	Actual	
p	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Create OplElection object		myElection is not null	myElection is not null	
2	Call runElection() method		runElectionTie() runs and doesn't throw any errors	runElectionTie() runs and doesn't throw any errors	
	Check that correct winners are declared	OPLTie.txt	The correct winners are declared	The correct winners are declared	

ı	Project Team# 5	Name:	Project	1:	Voting	System
	Test Stage: Unit	X Syster	n Tes	t Date: 5/	2/2021	
	Test Case ID#: OPL_TEST_TOOM	MANYSEATS_0	001 Nan	ne(s) of T	esters: Cole	
	Test Description:					
	Runs an OPL elect awarded more sea candidates. The se other parties.	ts than they ha	ve			
			(wh	at file) an	re are you storing d the name of the tions being used	e
			is st	ored in the	lectionTooManySe e TestOplElection hods from OplElec	class in src,
	Automated: yes_	<u>X</u> no	-			
	Results: Pass	<u>X</u> Fail				
•						
	Preconditions for Party.java, and Can- same directory as To	didate.java exi	st. The "OPLTOON	•	•	-

Ste			Expected	Actual	N-4
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Create OplElection object		myElection is not null	myElection is not null	
2	Call runLiection()		runElectionTooMan ySeats() runs and doesn't throw any errors	runElectionTooManySeats( ) runs and doesn't throw any errors	
3	Check that correct winners are declared	OPLTOOMAN YSEATS.txt	The correct winners are declared	The correct winners are declared	

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage: Unit	X System	Test	Date: 5/2	2/2021	
Test Case ID#: OPL_TEST_TOOM	MANYSEATS2_001	Nam	e(s) of Te	esters: Cole	
Test Description:					
	tion where there are in candidates and e led a seat.				
		(wha	at file) and	e are you storing If the name of the ions being used	<b>e</b>
		meth class	od is stor	ectionTooManySe ed in the TestOpII nd uses methods	Election
Automated: yes_	_X_ no				
Results: Pass	X Fail				
Preconditions for Party.java, and Can same directory as To	-	he "OPLTOOM	-	•	-

Ste		_	Expected Result	Actual	Notes
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1	Create OplElection object		myElection is not null	myElection is not null	
2	Call runElection() method	OPLTOOMAN	runElectionTooMan ySeats() runs and doesn't throw any errors	runElectionTooManySeats( ) runs and doesn't throw any errors	
3	Check that correct winners are declared	OPLTOOMAN YSEATS2.txt	The correct winners are declared	The correct winners are declared	

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage:	Unit X Syst	tem	Test Date: 5	5/2/2021	
Test Case ID	#: PARTY_TEST_	_001	Name(s) of	Testers: Cole	
Test Descrip	tion:				
	structor of the part iables are properly	•			
			(what file) ar	ere are you storing nd the name of the ctions being used.	9
				structor method is s ss in src and uses r	
Automated:	yes <u>X</u> no _				
Results: Pa	ss <u>X</u> Fail_				
Precondition	s for Test: JUnit c	ompiles properly	y and TestPar	ty.java, Party.java, e	exist.

Ste p #	Test Step Description	Test Data		Actual Result	Notes
1 4	Create Party object		Party name X is created	Party name X is created	

Project Name Team# 5	: Project	1:	Voting	System
Test Stage: Unit <u>X</u> S	System Tes	t Date: 5/	2/2021	
Test Case ID#: PARTY_TE	ST_002 Na	me(s) of Te	esters: Cole	
Test Description:				
Tests that addCandidate() m and that a candidate is prope				
	Ind	icate whei	e are you storing	g the tests
	-	-	d the name of the ions being used	
	the		andidate method i class in src and u	
Automated: yes <u>X</u> n	no			
Results: Pass X F	ail			
Preconditions for Test: JUr Candidate.java exist.	nit compiles properly an	d TestParty	.java, Party.java,	

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
			myParty is not null	myParty is not null	
1 4	Create Party object	char X			
2		Candidate(John,	properly adds the	addCandidate properly adds the candidate to the ArrayList	

The candidate was added to the ArrayList in Party.

Project Name: Team# 5	Project	1:	Voting	System
Test Stage: Unit <u>X</u> System _	Test	t Date: 5/2	2/2021	
Test Case ID#: PARTY_TEST_003	Nan	ne(s) of Te	esters: Cole	
Test Description:				
Tests that the rankCandidates metho works properly, and orders the candidorder based on their amount of votes	dates in			
	(wh	at file) and	e are you storing the name of the ions being used	9
	in th		Candidates metho y class in src and Party	
Automated: yes <u>X</u> no				
Results: Pass X Fail				
Preconditions for Test: JUnit compil Candidate.java exist.	les properly and	I TestParty	.java, Party.java,	

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
			myPartyn is not null	myParty is not null	
1	Create Party object	char X			
1 ^	Create Candidate objects and add them to Party object	X)	Candidates are properly added to the party	Candidates are properly added to the party	
3	Call rankCandidates method		The candidates are ranked in correct order	The candidates are ranked in the correct order.	

The candidates in the Party are ranked in the correct order according to their amount of ballots

Project Team# 5	Name:	Project	1:	Voting	System		
Test Stage: Unit	X System	. Test	Date: 5/2	2/2021			
Test Case ID#: PA	RTY_TEST_004	Nam	e(s) of Te	sters: Cole			
Test Description:							
1	Tests that the getter and setter for remainder works properly.						
		(wha	t file) and	e are you storing I the name of the ons being used	9		
		store		SetterRemainder estParty class in s Party			
Automated: yes_	<u>X</u> no						
Results: Pass <u>X</u>	( Fail						
Preconditions for Test: JUnit compiles properly and TestParty.java, Party.java exist.							

			Expected	Actual	
Ste p	Test Step	Test	Result	Result	Notes
#	Description	Data			
1 4	Create Party object	char X	myParty is not null	myParty is not null	
I ~	Call setRemainder and getRemainder		getRemainder	setRemainder and getRemainder properly work	

setRemainder and getRemainder properly work.

Project N Team# 5	lame:	Project	1:	Voting	System
Test Stage: Unit <u>X</u>	System	Test	Date: 5/2/	2021	
Test Case ID#: PAR	TY_TEST_005	Nam	e(s) of Tes	ters: Cole	
Test Description:					
Tests that addSeats a work.	nd addSeat meth	ods			
		lus all a			4b a 4a a4a
		(wha	t file) and	are you storing the name of the ons being used	9
		TestF		ts method is sto in src and uses	
Automated: yes X	no				
Results: Pass X	Fail	<u> </u>			
Preconditions for Tes	st: JUnit compiles	s properly and	TestParty.ja	ava, Party.java e	exist.

Cto			Expected	Actual	
Ste p	Test Step	Test	Result	Result	Notes
#	Description	Data			
			myParty is not null	myParty is not null	
1 4	Create Party object	char X			
1 ^	Call addSeats() and addSeat()		addSeat() both add	addSeats() and addSeat() both add the correct number of seats	

Both addSeats() and addSeat() add the correct number of seats to the total.

Project Team# 5	Name:	Project	1:	Voting	System
Test Stage: Unit	t <u>X</u> System _	_ Test	Date: 5/2	2/2021	
Test Case ID#: P	ARTY_TEST_006	Nan	ne(s) of Te	esters: Cole	
Test Description: Tests that assignS	: Seats() method work	s and			
1	winners for the party				
		(wh	at file) and	e are you storing the name of the ions being used	e
		Test	_	nSeats method is s in src and uses	
Automated: yes	_X no				
Results: Pass _	X Fail				
Preconditions for Candidate.java exis	· <b>Test:</b> JUnit compile st.	es properly and	l TestParty	.java, Party.java	

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
			myParty is not null	myParty is not null	
1	Create Party object	char X			
2	Create Candidate objects and add them to the party	Candidate(Jolly,	The candidates are properly created and added to the party	The candidates are properly created and added to the party	
3	Call the assignSeats() method		The correct candidate is awarded a seat	The correct candidate is awarded a seat	

The correct candidate was awarded the seat in the party.

Project Name: Team# 5	Project	1:	Voting	System
Test Stage: Unit X System	Test	Date: 5/	2/2021	
Test Case ID#: IR_TEST_CONSTRUCTOR_001	Nam	e(s) of Te	esters: loana	
Test Description:				
Tests that the runElection method in IrElection runs properly and correct the election.				
	(wha	t file) an	re are you storing d the name of the ions being used	9
	Testl	rElection	ructor method is s class in src, and u IrElection.	
Automated: yes_X_ no				
Results: Pass X Fail				
Preconditions for Test: JUnit comp Candidate.java exist. The "IR.txt" file			•	-

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1			Compiles with no errors	Compiles with no errors	
2		Use the Junit	Junit window should show a "Passed" message	Junit showed a "Passed" message	

Project Name: Team# 5	Project 1: Voting System
Test Stage: Unit X System	Test Date: 5/2/2021
Test Case ID#: IR_TEST_PROCESS_FILE_001	Name(s) of Testers: Ioana
Test Description:	
Tests that the runElection method in IrlElection runs properly and correctly ruthe election.	ns
	Indicate where are you storing the tests (what file) and the name of the method/functions being used.
	The testProcessOneFile method is stored in the TestIrElection class in src, and uses methods from IrElection.
Automated: yes <u>X</u> no	
Results: Pass <u>X</u> Fail	<del>-</del>
•	properly and TestIrElection.java, IrElection.java, and aced in the same directory as TestIrElection.java.

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1			Compiles with no errors	Compiles with no errors	
2		Use the Junit	Junit window should show a "Passed" message	Junit showed a "Passed" message	

Project Nam Team# 5	ne: Proj	ect 1	:	Voting	System
Test Stage: Unit <u>X</u>	System	Test Date	e: 5/2/202	1	
Test Case ID#: IR_TEST_ASSIGN_BALL	OTS_001	Name(s)	of Testers	s: loana	
Test Description:					
Tests that the runElection IrElection runs properly an the election.					
		(what file	e) and the	you storing name of the being used.	<b>,</b>
		the TestIr	-	ots method is ass in src, ar ction.	
Automated: yes_X_	no				
Results: Pass X	Fail				
Preconditions for Test: JU Candidate.java exist. The "I		-	-		-

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1			Compiles with no errors	Compiles with no errors	
2		Use the Junit	Junit window should show a "Passed" message	Junit showed a "Passed" message	

Project Na Team# 5	me: P	roject	1:	Voting	System
Test Stage: Unit <u>X</u>	System	Test	Date: 5/2	2021	
Test Case ID#: IR_TEST_RUN_ELECTI	ON_001	Nam	e(s) of Tes	sters: Ioana	
Test Description:					
Tests that the runElection method in IrElection runs properly and correctly runs the election.					
		(wha	t file) and	are you storing the name of the ons being used	9
		Testl		ction method is s lass in src, and u Election.	
Automated: yes <u>X</u>	no				
Results: Pass X	Fail				
<b>Preconditions for Test:</b> JUnit compiles properly and TestIrElection.java, IrElection.java, and Candidate.java exist. The "IR.txt" file is placed in the same directory as TestIrElection.java.					

Ste			Expected	Actual	
р	Test Step	Test	Result	Result	Notes
#	Description	Data			
1		• P • · ·	Compiles with no errors	Compiles with no errors	
2		framowork to run	Junit window should show a "Passed" message	Junit showed a "Passed" message	

Project Name: Team# 5	Project	1:	Voting	System
Test Stage: Unit <u>X</u> System	n Test	Date: 5/2	2/2021	
Test Case ID#: IR_TEST_RUN_ELECTION_2WAY	/TIE_001 Nam	e(s) of Te	esters: Ioana	
Test Description:				
Tests that the runElection method i IrElection runs properly and correcthe election.	tly runs India (wha metl	it file) and nod/funct testRunEl	e are you storing I the name of the ions being used ectionIR2WayTie	method is
			estIrElection class from IrElection.	s in src, and
Automated: yesX_ no				
Results: Pass <u>X</u> Fail				
Preconditions for Test: JUnit com Candidate.java exist. The "IR.txt" file			•	-

Ste p	Test Step	Test	•	Actual Result	Notes
F   #	Description	Data			
1			Compiles with no errors	Compiles with no errors	
2		Use the Junit	Junit window should show a "Passed" message	Junit showed a "Passed" message	

Project Name: Team# 5	Project	1:	Voting	System
Test Stage: Unit X System  Test Case ID#:  IR_TEST_ELIMINATE_BALLOT_001		Date: 5/	2/2021 esters: Ioana	
Test Description:  Tests that the testEliminateBallot metho IrElection runs properly and correctly rule the election.	-			
	(wha	t file) and	e are you storing d the name of the ions being used	9
	the T	estIrElect	ateBallot method ion class in src, a IrElection.	
Automated: yes <u>X</u> no				
Results: Pass X Fail	_			
Preconditions for Test: JUnit compiles Candidate.java exist. The "IR.txt" file is pl			•	•

Ste p	Test Step	Test	Expected Result	Actual Result	Notes
#	Description	Data			
1		1000	Compiles with no errors	Compiles with no errors	
2		framowork to run	Junit window should show a "Passed" message	Junit showed a "Passed" message	

Project Name: Team# 5	Project	1:	Voting	System
Test Stage: Unit X Sys	stem Tes	t Date: 5/	2/2021	
Test Case ID#: TEST_AUDIT_SERVICE_001	Nar	ne(s) of To	esters: Ioana	
Test Description:				
Tests that the runElection method in the election runs properly and conthe election.				
	(wh	at file) an	re are you storing d the name of the tions being used	e
	Tes	tIrElection	lection method is class in src, and u IrElection.	
Automated: yesX_ no				
Results: Pass X Fai	il			
<b>Preconditions for Test:</b> JUnit compiles properly and TestIrElection.java, IrElection.java, and Candidate.java exist. The "IR.txt" file is placed in the same directory as TestIrElection.java.				

Ste p	Test Step	Test	•	Actual Result	Notes
#	Description	Data			
1			Compiles with no errors	Compiles with no errors	
2		Use the Junit	Junit window should show a "Passed" message	Junit showed a "Passed" message	

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:	Test 1: PBI 5 Processing multiple files for IR
Team Member(s) Responsible:	Trevor
Inputs:	IR.txt and IR2.txt
Tests:	We run the election with multiple inputted files.
Outputs:	The GUI will display relevant information to the screen including number of ballots, votes received, and the winner. Both audit and media files are produced.
Passed or Failed	Pass
Date:	5/2/2021

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:	Test 2: PBI 5 Processing multiple files for OPL
Team Member(s) Responsible:	Cole
Inputs:	OPL.txt and OPL2.txt
Tests:	We run the election with multiple inputted files.
Outputs:	The GUI will display relevant information to the screen including number of ballots, votes received, and the winner. Both audit and media files are produced.
Passed or Failed	Pass
Date:	5/2/2021

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:	Test 3: PBI 7 Invalidating ballots for IR
Team Member(s) Responsible:	Ioana
Inputs:	IRInvalidBallots.txt
Tests:	We run the election with a file that contains ballots that should be marked as invalid.  We also run a JUnit test.
Outputs:	The GUI will display relevant information to the screen including number of ballots, votes received, and the winner. Both audit and media files are produced. It shows the number of ballots invalidated.  The JUnit framework gives a pass message.
Passed or Failed	Pass
Date:	5/2/2021

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:	Test 4: PBI 5 Allowing the system to process PO ballots.
Team Member(s) Responsible:	Trevor
Inputs:	PO.txt
Tests:	We run the election with a PO file. No information is printed to the screen. Therefore, much of this testing is done within the JUnit framework.
	testConstructor() testProcessOneFile() testMultipleFiles()
Outputs:	JUnit framework indicates a pass. The terminal gives no errors.
Passed or Failed	Pass
Date:	5/2/2021