Project Name: Voting System	Team# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: CP1 Test Description: Tests the CPLParty Constructor	Name(s) of Testers: Grant Oie
setup with: std::vector <candidate*> candidates; candidates.push_back(new CPLCandidate("Candidate 1")); candidates.push_back(new CPLCandidate("Candidate 2")); candidates.push_back(new CPLCandidate("Candidate 3"));</candidate*>	
	Filename: CPLPartyUnitTest.cpp
	Testname: CPLPartyTest, ConstructorTest
Automated: yes X no	Functions: CPLParty()
Results: Pass X Fail	
Preconditions for Test: Accurate types passed into constructo	r

Step #	Test Step Description	Test Data	•	Actual Result	Notes
1	_	party candidates			
2	test party name	->get name()	Party 1	Party 1	
3	get the candidates vect	party->get candidates()			
4	check size of each vector		3	3	
5	check equality of Candidate objects		all objects equal	all objects equal	

Post condition(s) for Test: party is properly instantiated

Team# 13
Test Date: 3/24/24
Name(s) of Testers: Grant Oie
Filename: CPLPartyUnitTest.cpp Testname: CPLPartyTest, GetCandidatesTest Functions: get_candidates(), get_name

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	3	party			
1	object	candidates			
		partyCandidates =			
2	get the candidates vect	party->get candidates()			
	check size of candidates		3	3	
3	vectors				
	Check that candidate in party		Candidate 1	Candidate 1	
4	matches name of "Candidate 1"	partyCandidates			
	Check that candidate in party		Candidate 2	Candidate 2	
5	matches name of "Candidate 2"	partyCandidates			
	Check that candidate in party		Candidate 3	Candidate 3	
6	matches name of "Candidate 3"	partyCandidates			

Post condition(s) for Test: get_candidates properly returns candidate vector

Project Name: Voting System	Team# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: CP3 Test Description: Tests get_name()	Name(s) of Testers: Grant Oie
setup with: std::vector <candidate*> candidates; candidates.push_back(new CPLCandidate("Candidate 1")); candidates.push_back(new CPLCandidate("Candidate 2")); candidates.push_back(new CPLCandidate("Candidate 3"));</candidate*>	
	Filename: CPLPartyUnitTest.cpp
	Testname: CPLPartyTest, GetNameTest
	Functions: get_name()
Automated: yes X no	
Results: Pass X Fail	
Preconditions for Test:	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1		party candidates			
2	test party name	->get name()	Party 1	Party 1	

Project Name: Voting System	leam# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: CP4 Test Description: Tests the set_total_votes and get_total_votes function	Name(s) of Testers: Grant Oie
setup with: std::vector <candidate*> candidates; candidates.push_back(new CPLCandidate("Candidate 1")); candidates.push_back(new CPLCandidate("Candidate 2")); candidates.push_back(new CPLCandidate("Candidate 3"));</candidate*>	
	Filename: CPLPartyUnitTest.cpp
	Testname: CPLPartyTest, GetSetTotalVotes
	Functions: get_total_votes, set_total_votes
Automated: yes X no	
Results: Pass X Fail	
Preconditions for Test:	
2.1000.1410.1010 202 2000	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
	,	party candidates			
2	set total votes to 5	party			
3	get total votes, check equal to 5	party	5	5	
4					
5					

Post condition(s) for Test: party is properly instantiated

Project Name: Voting System	Team# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: CP5 Test Description: Tests the assignseatwinner on party w/ 3 candidates, 2 seats	Name(s) of Testers: Grant Oie
setup with: std::vector <candidate*> candidates; candidates.push_back(new CPLCandidate("Candidate 1")); candidates.push_back(new CPLCandidate("Candidate 2")); candidates.push_back(new CPLCandidate("Candidate 3"));</candidate*>	
	Filename: CPLPartyUnitTest.cpp Testname: CPLPartyTest, AssignSeatWinners_base Functions: assign_seat_winners(seats)
Automated: yes X no	
Results: Pass X Fail	
Preconditions for Test: seats parameter >= 0	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	_	party candidates			
2	call assign seatwinners(2)				
	check that first two candidates got winner designation		true	true	

Project Name: Voting System	Team# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: CP6 Test Description: Tests the assignseatwinner on party w/ 3 candidates, 0 seats	Name(s) of Testers: Grant Oie
setup with: std::vector <candidate*> candidates; candidates.push_back(new CPLCandidate("Candidate 1")); candidates.push_back(new CPLCandidate("Candidate 2")); candidates.push_back(new CPLCandidate("Candidate 3"));</candidate*>	
	Filename: CPLPartyUnitTest.cpp Testname: CPLPartyTest, AssignSeatWinners_zero Functions: assign_seat_winners(seats)
Automated: yes X no	
Results: Pass X Fail	
Preconditions for Test: seats parameter >= 0	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1 1		party candidates			
2	call assign seatwinners(0)				
1	check that no candidates were assigned seat		true	true	

Project Name: Voting System	Team# 13		
Test Stage: Unit _X_ System	Test Date: 3/24/24		
Test Case ID#: CP7 Test Description: Tests the assignseatwinner on party w/ 3 candidates, 4 seats	Name(s) of Testers: Grant Oie		
setup with: std::vector <candidate*> candidates; candidates.push_back(new CPLCandidate("Candidate 1")); candidates.push_back(new CPLCandidate("Candidate 2")); candidates.push_back(new CPLCandidate("Candidate 3"));</candidate*>			
Automated: yes X no	Filename: CPLPartyUnitTest.cpp Testname: CPLPartyTest, AssignSeatWinners_over Functions: assign_seat_winners(seats)		
Results: Pass X Fail			
1411			
Preconditions for Test: seats parameter >= 0			

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
	1	party candidates			
2	call assign seatwinners(4)				
	check that all candidates were assigned seat		true	true	

Project Name: Voting System	Team# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: OP1 Test Description: Tests the OPLParty Constructor	Name(s) of Testers: Grant Oie
setup with: std::vector <candidate*> candidates; candidates.push_back(new OPLCandidate("Candidate 1")); candidates.push_back(new OPLCandidate("Candidate 2")); candidates.push_back(new OPLCandidate("Candidate 3"));</candidate*>	
Automated: yes X no	Filename: FileOPLPartyUnitTest.cpp Testname: OPLPartyTest, ConstructorTest Functions: OPLParty()
Results: Pass X Fail	

Preconditi	s for Test: Accurate types passed into constructor	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
	-	party candidates			
2	test party name	->get name()	Party 1	Party 1	
3	get the candidates vect	party->get candidates()			
4	check size of each vector		3	3	
	check equality of Candidate objects		all objects equal	all objects equal	

Post condition(s) for Test: party is properly instantiated

Project Name: Voting System	Team# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: OP2 Test Description: Tests the get_candidates function	Name(s) of Testers: Grant Oie
setup with: std::vector <candidate*> candidates; candidates.push_back(new OPLCandidate("Candidate 1")); candidates.push_back(new OPLCandidate("Candidate 2")); candidates.push_back(new OPLCandidate("Candidate 3"));</candidate*>	
Automated: yes X no	Filename: FileOPLPartyUnitTest.cpp Testname: OPLPartyTest, GetCandidatesTest Functions: get_candidates(), get_name
Results: Pass X Fail	
Preconditions for Test:	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	instantiate new OPLParty	party			
1	object	candidates			
		partyCandidates =			
2	get the candidates vect	party->get candidates()			
	check size of candidates		3	3	
3	vectors				
	Check that candidate in party		Candidate 1	Candidate 1	
4	matches name of "Candidate 1"	partyCandidates			
	Check that candidate in party		Candidate 2	Candidate 2	
5	matches name of "Candidate 2"	partyCandidates			
	Check that candidate in party		Candidate 3	Candidate 3	
6	matches name of "Candidate 3"	partyCandidates			

Post condition(s) for Test: get_candidates properly returns candidate vector

Project Name: Voting System	Team# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: OP3 Test Description: Tests get_name()	Name(s) of Testers: Grant Oie
setup with: std::vector <candidate*> candidates; candidates.push_back(new OPLCandidate("Candidate 1")); candidates.push_back(new OPLCandidate("Candidate 2")); candidates.push_back(new OPLCandidate("Candidate 3"));</candidate*>	
	Filename: FileOPLPartyUnitTest.cpp Testname: OPLPartyTest, GetNameTest Functions: get_name()
Automated: yes X no	
Results: Pass X Fail	
Preconditions for Test:	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
	_	party candidates			
2	test party name	->get name()	Party 1	Party 1	

Project Name: Voting System	Team# 13		
Test Stage: Unit _X_ System	Test Date: 3/24/24		
Test Case ID#: OP4 Test Description: Tests the set_total_votes and get_total_votes function	Name(s) of Testers: Grant Oie		
setup with: std::vector <candidate*> candidates; candidates.push_back(new OPLCandidate("Candidate 1")); candidates.push_back(new OPLCandidate("Candidate 2")); candidates.push_back(new OPLCandidate("Candidate 3"));</candidate*>			
	Filename: FileOPLPartyUnitTest.cpp		
	Testname: OPLPartyTest, GetSetTotalVotes		
	Functions: get_total_votes, set_total_votes		
Automated: yes X no			
Results: Pass X Fail			
Preconditions for Test:			

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
		party candidates			
2	set total votes to 5	party			
3	get total votes, check equal to 5	party	5	5	
4					
5					

party is properly instantiated

Team# 13 Project Name: Voting System Test Stage: Unit X **Test Date: 3/24/24** System **Test Case ID#: OP5** Name(s) of Testers: Grant Oie **Test Description:** Tests the num votes attribute, and calculate total votes, get total votes functions calculate total votes is called in constructor and not immediately visible setup with: std::vector<Candidate*> candidates; candidates.push back(new OPLCandidate("Candidate 1")); candidates.push back(new OPLCandidate("Candidate 2")); candidates.push back(new OPLCandidate("Candidate 3")); Filename: FileOPLPartyUnitTest.cpp **Testname:** OPLPartyTest, GetTotalVotesTest Functions: ->num votes, get total votes, calculate total votes Automated: yes X no **Results: Pass** Fail **Preconditions for Test:**

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Assign num_votes to candidates in candidates vector	candidates			
	-	party candidates			
	check that party.get_total_votes == sum of votes assigned to candidates		60	60	

Project Name: Voting System	Team# 13		
Test Stage: Unit _X_ System	Test Date: 3/24/24		
Test Case ID#: OP6	Name(s) of Testers: Grant Oie		
Test Description: Tests assign_seat_winners with 1 seat and a clear winner and get_winner functions			
setup with: std::vector <candidate*> candidates; candidates.push_back(new OPLCandidate("Candidate 1")); candidates.push_back(new OPLCandidate("Candidate 2")); candidates.push_back(new OPLCandidate("Candidate 3"));</candidate*>			
	Filename: FileOPLPartyUnitTest.cpp Testname: OPLPartyTest,AssignSeatWinners_singleSeat Functions: assign_seat_winners, get_winner		
Automated: yes X no			
Results: Pass X Fail			

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Assign num_votes to candidates in candidates vector: 30, 20, 10 respectively	candidates			
2	instantiate new OPLParty	party candidates			
3	call party->assign_seat_winners(1)	party			
4	Assert_true(candidate[0]->get_winner())		true	true	
5	Assert_true(candidate[1]->get_winner())		false	false	
6	Assert_true(candidate[2]->get_winner())		false	false	

Preconditions for Test: XXX

Project Name: Voting System	Team# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: OP7	Name(s) of Testers: Grant Oie
Test Description: Tests assign_seat_winners with 2 seat and a clear winners and get_winner functions	
setup with: std::vector <candidate*> candidates; candidates.push_back(new OPLCandidate("Candidate 1")); candidates.push_back(new OPLCandidate("Candidate 2")); candidates.push_back(new OPLCandidate("Candidate 3"));</candidate*>	
• • • • • • • • • • • • • • • • • • •	Filename: FileOPLPartyUnitTest.cpp
	Testname: OPLPartyTest,AssignSeatWinners_multipleSeats
Automated: yes X no	Functions: assign_seat_winners, get_winner
Results: Pass X Fail	
Preconditions for Test: XXX	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Assign num_votes to candidates in candidates vector: 30, 25, 20 respectively	candidates			
2	,	party candidates			
3	call party->assign_seat_winners(2)	party			
4	Assert_true(candidate[0]->get_winner())		true	true	
5	Assert_true(candidate[1]->get_ winner())		true	true	
6	Assert_true(candidate[2]->get_winner())		false	false	

Project Name: Voting System	Team# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: OP8 Test Description: Tests the assign_seat_winners() function under a tie-breaker scenario. 3-way tie for 2 seats. Ran several times to verify that the same candidates are not being selected each time.	Name(s) of Testers: Grant Oie
setup with: std::vector <candidate*> candidates; candidates.push_back(new OPLCandidate("Candidate 1")); candidates.push_back(new OPLCandidate("Candidate 2")); candidates.push_back(new OPLCandidate("Candidate 3"));</candidate*>	
Automated: yes X no	Filename: FileOPLPartyUnitTest.cpp Testname: OPLPartyTest, AssignSeatWinners_equalVotes Functions: assign_seat_winner
Results: Pass X Fail	
Preconditions for Test:	

Step	-	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	Assign num_votes to candidates in candidates vector: 30 for all	candidates			
	-	party candidates			
	call party->assign seat winners(2)	party			
4	sum number of winners in				

	candidate array			
5	check num winners	2	2	

Project Name: Voting System Team# 13

Test Stage: Unit _X_ System __ Test Date: 3/24/24

Test Case ID#: EDP1 Name(s) of Testers: Grant Oie

Test Description:

test tokenize lines against, base case

setup with:

oplTestFile = "../testing/test_data/sys_test1_opl.csv";

cplTestFile = "../testing/test data/sys test3 cpl.csv";

std::string oplTestFile;

std::string cplTestFile;

std::string election_type;

int numSeats;

int numBallots;

std::string line;

int numCandidates;

int numParties;

Filename: ElectionDataParserUnitTest.cpp

Testname: ElectionDataParserTest, TokenizeLines base

Functions: tokenize lines

Automated: yes_X_ no

Results: Pass X Fail

Preconditions for Test:

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	initialize string	"Token1,Token2,Token3"			
2	call tokenize lines on string	vector <string></string>			
3	check size of vector		3	3	
4	check equality of vector[0]		"Token1"	"Token1"	
5	check equality of vector[1]		"Token2"	"Token2"	
6	check equality of vector[2]		"Token3"	"Token3"	

Project Name: Voting System Team# 13

Test Stage: Unit _X_ System __ Test Date: 3/24/24

Test Case ID#: EDP2 Name(s) of Testers: Grant Oie

Test Description:

test tokenize_lines with whitespaces added

setup with:

oplTestFile = "../testing/test_data/sys_test1_opl.csv";

cplTestFile = "../testing/test_data/sys_test3_cpl.csv";

std::string oplTestFile;

std::string cplTestFile;

std::string election_type;

int numSeats;

int numBallots;

std::string line;

int numCandidates;

int numParties;

Filename: ElectionDataParserUnitTest.cpp

Testname: ElectionDataParserTest, TokenizeLines_whitespaces

Functions: tokenize lines

Automated: yes_X_ no_

Results: Pass X Fail

Preconditions for Test:

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	initialize string	"Token1, Token2,Token3"			
2	call tokenize_lines on string	vector <string></string>			
3	check size of vector		3	3	
4	check equality of vector[0]		"Token1"	"Token1"	
5	check equality of vector[1]		"Token2"	"Token2"	
6	check equality of vector[2]		"Token3"	"Token3"	

Project Name: Voting System

Team# 13

Test Stage: Unit _X_ System __ Test Date: 3/24/24

Test Case ID#: EDP3 Name(s) of Testers: Grant Oie

Test Description:

test tokenize lines on a ballot example

setup with:

oplTestFile = "../testing/test_data/sys_test1_opl.csv";

cplTestFile = "../testing/test data/sys test3 cpl.csv";

std::string oplTestFile;

std::string cplTestFile;

std::string election_type;

int numSeats;

int numBallots;

std::string line;

int numCandidates;

int numParties;

Filename: ElectionDataParserUnitTest.cpp

Testname: ElectionDataParserTest, TokenizeLines_ballots1

Functions: tokenize_lines

Automated: yes X no

Results:	Pass _	_X_	Fail_	
Precondi	tions for	r Test:		

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	initialize string	",,1, "			
2	call tokenize_lines on string	vector <string></string>			
3	check size of vector		4	4	
4	check equality of vector[0]		((?)	(())	
5	check equality of vector[1]		(())	(67)	
6	check equality of vector[2]		"1"	"1"	
7	check equality of vector[3]		((?)	(())	

int numParties;

Project Name: Voting System Team# 13

Test Stage: Unit _X_ System __

Test Case ID#: EDP4

Test Description:

test tokenize_lines on a ballot example

setup with:

oplTestFile = "../testing/test_data/sys_test1_opl.csv";

cplTestFile = "../testing/test_data/sys_test3_cpl.csv";

std::string oplTestFile;

std::string cplTestFile;

std::string election_type;

int numSeats;

int numBallots;

std::string line;

int numCandidates;

Test Date: 3/24/24

Name(s) of Testers: Grant Oie

Preconditions for Test:	
Results: Pass X Fail	
Automated: yes X no	_
	Filename: ElectionDataParserUnitTest.cpp Testname: ElectionDataParserTest, TokenizeLines_ballots2 Functions: tokenize_lines

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	initialize string	"1,,,"			
2	call tokenize lines on string	vector <string></string>			
3	check size of vector		4	4	
4	check equality of vector[0]		"1"	"1"	
5	check equality of vector[1]		((?)	(())	
6	check equality of vector[2]		((2)	(())	
7	check equality of vector[3]		(6)	607	

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	initialize string	",,,1 "			
2	call tokenize lines on string	vector <string></string>			
3	check size of vector		4	4	
4	check equality of vector[0]		((2)	(67)	
5	check equality of vector[1]		((2)	(6)	
6	check equality of vector[2]		((2)	(0)	
7	check equality of vector[3]		"1"	"1"	

```
Team# 13
Project Name: Voting System
Test Stage: Unit X
                                                             Test Date: 3/24/24
                           System
Test Case ID#: EDP6
                                                             Name(s) of Testers: Grant Oie
Test Description:
test the accuracy of create OPL candidates function
setup with:
oplTestFile = "../testing/test data/sys test1 opl.csv";
cplTestFile = "../testing/test data/sys test3 cpl.csv";
std::string oplTestFile;
std::string cplTestFile;
std::string election type;
int numSeats;
int numBallots;
std::string line;
int numCandidates;
int numParties;
                                                               Filename: ElectionDataParserUnitTest.cpp
                                                               Testname: ElectionDataParserTest, CreateOPLCandidates
                                                               Functions: create OPL candidates
Automated: yes X no
Results: Pass X
                         Fail
Preconditions for Test: there are candidates in the csv file
```

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	open opltestfile				
2	check that file opened properly				
3	read first line into electionType	electionType			
4	read second line into numSeats	numSeats			
5	read third line into numBallots	numBallots			
6	read fourth line into numCandidates	numCandidates			
7	call create_OPL_candidates(file,	candidates = vector <tuple<partyname, candidate*>></tuple<partyname, 			
8	check equality of numSeats, numBallots, numCandidates, candidates.size() against data in file		true	true	
9	check each partyname and candidate's name in the candidates vector, calling ->get_name() on each candidate and expect_eq'ing against the relevant name in the csv file		true	true	
	delete allocated candidate memory				

Team# 13

Post condition(s) for Test:

Project Name: Voting System

Test Stage: Unit _X_ System __ Test Date: 3/24/24

Test Case ID#: EDP7 Name(s) of Testers: Grant Oie

Test Description:

test the accuracy of assign_votes_to_candidates function

setup with:

oplTestFile = "../testing/test data/sys test1 opl.csv";

cplTestFile = "../testing/test_data/sys_test3_cpl.csv";

std::string oplTestFile; std::string cplTestFile;

std::string election_type;
int numSeats;
int numBallots;
std::string line;
int numCandidates;
int numParties;

Filename: ElectionDataParserUnitTest.cpp

Testname: ElectionDataParserTest, AssignVotesToCandidates **Functions:** create_OPL_candidates,assign_votes_to_candidates

P	lut	tomat	<u>ted</u>	: :	yes	X	no

Results:	Pass	\mathbf{X}	Fail
----------	------	--------------	------

Preconditions for Test: there are candidates in the csv file, there are ballots in the csv file

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	open opltestfile				
	check that file opened properly				
3	read first line into electionType	electionType			
4	read second line into numSeats	numSeats			
5	read third line into numBallots	numBallots			
6	read fourth line into numCandidates	numCandidates			
7	call create_OPL_candidates(file, numCandidates)	candidates = vector <tuple<partyname, candidate*>></tuple<partyname, 			
8	extract just candidates from the vector <tuple>></tuple>	candidatesVec			
9	call assign_votes_to_candidates(fil e,candidatesVec)				
10	close file				
11	check equality of candidates[i]->get_num_votes() against the expected vote value		true	true	
12	delete allocated candidate memory				

Project Name: Voting System	Team# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: EDP8 Test Description: test the accuracy of create_opl_parties function	Name(s) of Testers: Grant Oie
setup with: oplTestFile = "/testing/test_data/sys_test1_opl.csv"; cplTestFile = "/testing/test_data/sys_test3_cpl.csv"; std::string oplTestFile; std::string cplTestFile; std::string election_type; int numSeats; int numBallots; std::string line; int numCandidates; int numParties;	
	Filename: ElectionDataParserUnitTest.cpp Testname: ElectionDataParserTest, CreateOPLParties Functions: create_OPL_candidates,create_OPL_Parties
Automated: yes_X_ no	
Preconditions for Test: there are candidates in the csv fil	le

Step #	Test Step Description	Test Data	1	Actual Result	Notes
1	open opltestfile				
2	check that file opened properly				

2	16 11 1 1 1 7	1 7			
3	read first line into electionType	election lype			
4	read second line into numSeats	numSeats			
5	read third line into numBallots	numBallots			
6	read fourth line into numCandidates	numCandidates			
7	create_OPL_candidates(file,	candidates = vector <tuple<partyname, candidate*>></tuple<partyname, 			
8	call create_OPL_parties(candidates)	parties			
9	call assign_votes_to_candidates(fil e,candidatesVec)				
10	check parties size		3	3	
	check parties name and size of candidate vector against expected values		true	true	
	delete allocated party and candidate memory				

Project Name: Voting System Team# 13

Test Stage: Unit _X_ System __ Test Date: 3/24/24

Test Case ID#: EDP9
Name(s) of Testers: Grant Oie
Test Description:

test the accuracy of create_cpl_parties function

setup with:

oplTestFile = "../testing/test_data/sys_test1_opl.csv";

cplTestFile = "../testing/test_data/sys_test3_cpl.csv";

std::string oplTestFile;

std::string cplTestFile;

std::string election_type;

int numSeats;

int numBallots;

std::string line;

int numCandidates;

int numParties;

			Filename: ElectionDataParserUnitTest.cpp Testname: ElectionDataParserTest, CreateCPLParties Functions:create_CPL_Parties				
Automated: yes	<u>X</u> _	no					
Results: Pass	X	Fail					
Preconditions for Test: there are parties/candidates in the csv file							

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	open cpltestfile				
2	check that file opened properly				
3	read first line into electionType	electionType			
4	read second line into numSeats	numSeats			
5	read third line into numBallots	numBallots			
6	read fourth line into numCandidates	numCandidates			
7	call create_CPL_parties(file, numCandidates)	parties = vector <party*></party*>			
8	close file				
9	check parties size		6	6	
	check party name and size of candidate array against expected values		true	true	
11	delete allocated party memory				

Project Name: Voting System	Team# 13
Test Stage: Unit _X_ System	Test Date: 3/24/24
Test Case ID#: EDP10	Name(s) of Testers: Grant Oie
Test Description:	

test the accuracy of assign_votes_to_parties function

setup with:

oplTestFile = "../testing/test_data/sys_test1_opl.csv";

cplTestFile = "../testing/test_data/sys_test3_cpl.csv";

std::string oplTestFile;

std::string cplTestFile;

std::string election_type;

int numSeats;

int numBallots;

std::string line;

int numCandidates;

int numParties;

Filename: ElectionDataParserUnitTest.cpp

Testname: ElectionDataParserTest, AssignVotesToParties **Functions:**create CPL parties, assign votes to parties

Automated: yes X no

Results: Pass X Fail

Preconditions for Test: there are parties and votes in the csv file

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
	_				
1	open cpltestfile				
2	check that file opened properly				
3	read first line into electionType	electionType			
4	read second line into numSeats	numSeats			
5	read third line into numBallots	numBallots			
6	read fourth line into numCandidates	numCandidates			
7	call create_CPL_parties(file, numCandidates)	parties = vector <party*></party*>			
8	call assign_votes_to_parties(file,par ties)				
9	close file				
10	check parties vote counts		true	true	

	against expected values		
11	delete allocated party memory		

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit X System	Test Date: 22/03/2024
Test Case ID#: OC 1 Test Description: Tests OPLCandidate get_name() return value.	Name(s) of Testers: Michael Mulhall
Automated: yes X no	Repo-Team13/Project1/testing/OPLCandidateUnitTest.cpp
Results: Pass X Fail	
Preconditions for Test: OPLCandidate compiles and a Cand	idate object is initialized correctly.

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
2		OPLCandidate* joe = new OPLCandidate("Joe Schmo")	None	None	Candidate joe is initialized
3	Compare get_name to expected result	EXPECT_EQ("Joe Schmo", joe->get_name());	True		joe->get_name() returns Joe Schmo
		OPLCandidate* sam = new OPLCandidate("Sam	None	None	Candidate sam is initialized
4	-	Politician")			
5	Compare get_name to expected	_ <	True		sam->get_name() returns Sam Politician

D 4	1.4.	΄ \	r	Tr 4
Post	condition(S)	Ior	rest:

Get_name correctly outputs a Candidate's name. It can be used in the to_string function.

Project Name: Project 1: Voting System

Test Stage: Unit X System __ Test Date: 22/03/2024

Test Case ID#: OC 2 Name(s) of Testers: Michael Mulhall

Test Description: Tests OPLCandidate get_winner() return

value.

repo-Team13/Project1/testing/OPLCandidateUnitTest.cpp

Team#13

Automated: yes X no

Results: Pass X Fail

Preconditions for Test: OPLCandidate compiles and a Candidate object is initialized correctly.

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
2		OPLCandidate* steven = new OPLCandidate ("Steven Carter");	Candidate initialized	Candidate initialzed	Candidate initialized
3	Compare winner variable with default winner value of false	EXPECT_EQ(false, steven->get winner());	True	True	Candidate initialized with a winner value of false
4	Set winner to true	steven->set winner(true);	Winner set to true	Winner set to true	
5		EXPECT_EQ(true, steven->get winner());	True		Get_winner returns correct value of winner

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit X System	Test Date: 22/03/2024
Test Case ID#: OC 3 Test Description: Tests OPLCandidate setWinnerTest () return value.	Name(s) of Testers: Michael Mulhall
Automated: yes X no	Repo-Team13/Project1/testing/OPLCandidateUnitTest.cpp
Results: Pass X Fail	
Preconditions for Test: OPLCandidate compiles and a Car	ndidate object is initialized correctly
1 reconditions for rest. Of Deandidate compiles and a Car	ididate object is initialized correctly.

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
2		OPLCandidate* jenna = new OPLCandidate("Jenna America");			
3	Default get_winner value is checked	EXPECT_EQ(false, jenna->get winner());	True	True	
4		jenna->set_winner(true); EXPECT_EQ(true, jenna->get_winner());	True	True	
5			True	True	

Set_winner correctly sets the winner status to the parameter that is entered.

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit X System	Test Date: 22/03/2024
Test Case ID#: OC 4 Test Description: Tests OPLCandidate toStringTest() return value.	Name(s) of Testers: Michael Mulhall
Automated: yes X no	Repo-Team13/Project1/testing/OPLCandidateUnitTest.cpp
Results: Pass X Fail	
Preconditions for Test: OPLCandidate compiles and a Candidate	date object is initialized correctly.

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
2		OPLCandidate* tom = new OPLCandidate("Tom Clancy");			
3	Tom's votes set to 5	Tom->set num votes(5);	Num_votes value set to 5	Num_votes value set to 5	
4		EXPECT_EQ("5 Tom Clancy", tom->to string());	True	True	Using winner default value of 5
5	Tom set winner(true)	Tom->set winner(true);	Winner = true	Winner = true	
6	Tom to_string expected to return "5 Tom Clancy	EXPECT_EQ("> 5 Tom Clancy - WINNER", tom->to_string()); EXPECT_NE("5 Tom Clancy", tom->to_string());	True	True	To_string returns the correct values

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit X System	Test Date: 22/03/2024
Test Case ID#: OC 5 Test Description: Tests OPLCandidate getandsetNumVotesTest() return value/ altered value.	Name(s) of Testers: Michael Mulhall
Automated: yes X no	Repo-Team13/Project1/testing/OPLCandidateUnitTest.cpp
Results: Pass X Fail	
Preconditions for Test: OPLCandidate compiles and a Ca	indidate object is initialized correctly.

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
2		OPLCandidate("Jerry Seinfeld");	Jerry initialized	Jerry initialized	
3	Testing Jerry's expected default num votes value of zero	EXPECT_EQ(0, jerry->get num votes());	True	True	Get_num_votes() returns correct default value of 0
4	Jerry's votes set to 6	jerry->set num votes(6);	Num_votes set to 5	Num_votes set to 5	Using winner default value of 5
5		EXPECT_EQ(6, jerry->get num votes());	True	True	
6	Set Jerry's votes to new value 0	jerry->set_num_votes(0);	Num_votes set to 0	Num_votes set to 0	
7		EXPECT_EQ(0, jerry->get num votes());	True		Set_votes correctly alters the num votes value

Project Name: Project 1: Voting System Team#13

Test Stage: Unit X System __ Test Date: 22/03/2024

Test Case ID#: CC 1 Name(s) of Testers: Michael Mulhall

Test Description: Tests CPLCandidate get name() return

value.

Repo-Team13/Project1/testing/CPLCandidateUnitTest.cpp

Automated: yes X no

Results: Pass X Fail

Preconditions for Test: CPLCandidate compiles and a Candidate object is initialized correctly.

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
2		CPLCandidate* joe = new CPLCandidate("Joe Schmo")	None	None	Candidate joe is initialized
	Compare get_name to expected result	EXPECT_EQ("Joe Schmo", joe->get_name());	True		joe->get_name() returns Joe Schmo
4		CPLCandidate* sam = new CPLCandidate("Sam Politician")	None	None	Candidate sam is initialized
_	Compare get_name to expected	(True		sam->get_name() returns Sam Politician

Get_name correctly outputs a Candidate's name. It can be used in the to_string function.

Project Name: Project 1: Voting System	Team#13					
Test Stage: Unit X System	Test Date: 22/03/2024					
Test Case ID#: CC 2 Test Description: Tests CPLCandidate get_winner() return value.	Name(s) of Testers: Michael Mulhall					
Automated: yes X no	repo-Team13/Project1/testing/CPLCandidateUnitTest.cpp					
Results: Pass X Fail						
Preconditions for Test: CPLCandidate compiles and a CPLCandidate object is initialized correctly.						

Step #	Test Step Description	Test Data	_ <u> </u>	Actual Result	Notes
1	1				
2	Candidate steven initialized	CPLCandidate* steven = new CPLCandidate ("Steven Carter");	Candidate initialized	Candidate initialzed	Candidate initialized
3	Compare winner variable with default winner value of false	EXPECT_EQ(false, steven->get winner());	True	True	Candidate initialized with a winner value of false
4	Set winner to true	steven->set winner(true);	Winner set to true	Winner set to true	
5		EXPECT_EQ(true, steven->get_winner());	True	True	Get_winner returns correct value of winner

Post condition(s)	for	Test:
-----------------	----	-----	-------

Project Name: Project 1: Voting System	leam#13			
Test Stage: Unit X System	Test Date: 22/03/2024			
Test Case ID#: CC 3 Test Description: Tests CPLCandidate set_winner() return value.	Name(s) of Testers: Michael Mulhall			
Automated: yes X no	Repo-Team13/Project1/testing/CPLCandidateUnitTest.cpp			
Results: Pass X Fail				
Preconditions for Test: CPLCandidate compiles and a Candidate object is initialized correctly.				

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
2	OPLCandidate jenna initialized	CPLCandidate* jenna = new CPLCandidate("Jenna America");			
	Default get_winner value is checked	EXPECT_EQ(false, jenna->get winner());	True	True	
4	Jenna set to winner as true	jenna->set_winner(true); EXPECT_EQ(true, jenna->get_winner());	True	True	
5	Jenna set to winner as false		True	True	
	perma sec to winner as raise	germa - get_willier()),			

Set_winner correctly sets the winner status to the parameter that is entered.

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit X System	Test Date: 22/03/2024
Test Case ID#: CC 4 Test Description: Tests CPLCandidate toStringTest() return value.	Name(s) of Testers: Michael Mulhall
Automated: yes X no	Repo-Team13/Project1/testing/CPLCandidateUnitTest.cpp
Results: Pass X Fail	
Preconditions for Test: CPLCandidate compiles and a Candid	date object is initialized correctly.

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
2	CPLCandidate tom initialized	CPLCandidate* tom = new CPLCandidate("Tom Clancy");			
3	Tom's votes set to 5	Tom->set num votes(5);	Num_votes value set to 5	Num_votes value set to 5	
	Expect to_string to output "5 Tom Clancy"	EXPECT_EQ("5 Tom Clancy", tom->to string());	True	True	Using winner default value of 5
5	Tom set_winner(true)	Tom->set_winner(true);	Winner = true	Winner = true	
		EXPECT_EQ("> 5 Tom Clancy - WINNER", tom->to_string()); EXPECT_NE("5 Tom Clancy", tom->to_string());	True		To_string returns the correct values

P	ost	condition	(s) for	Test	t:
---	-----	-----------	----	-------	------	----

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit X System	Test Date: 22/03/2024
Test Case ID#: E1 Test Description: Tests ElectionData break_tie return value.	Name(s) of Testers: Michael Mulhall
Automated: yes X no	Repo-Team13/Project1/testing/ElectionDataUnitTest.cpp
Results: Pass X Fail	
Preconditions for Test: ElectionData object is initialized. Brea	ik tie is between at least two candidates.

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
	Integer num1 to be set equal to break tie(5)		Num1 should be a random integer in a range of 0 to 4		
3		EXPECT_GE(num1, 0); EXPECT_LE(num1, 4);	True	True	
4	Expect num2 to be set equal to break tie(72)		Num2 should be a random integer between 0 and 71		
		EXPECT_GE(num2, 0); EXPECT_LE(num2, 71);	True	True	
	Integer num3 to be set equal to break tie(2)		Num3 should be a random integer between 0 and 1 inclusive.		
		EXPECT_GE(num3, 0); EXPECT_LE(num3, 1);	True	True	

Post condition(s)	for Test:
-------------------	-----------

Project Name: Project 1: Voting System	Team# 13
Test Stage: Unit _x_ System	Test Date: March 22nd
Test Case ID#: AL1 Test Description:	Name(s) of Testers: Khalid Qasim
This test verifies the Audit Log constructor initializes the log with an empty string	
	Filename: AuditLogUnitTest.cpp
	Testname: AuditLogTest, ConstructorTest
	Functions: AuditLog()
Automated: yes X no	
Results: Pass X Fail	
Preconditions for Test: Audit Log object instantiated and t	est directory exists

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
	Call write_to_file method	l	N/A	N/A	
1	without adding any content to	Filename =			
1	the log	"ConstructorTest.txt"			
		testDirectory and Filename =	N/A	N/A	
2	Construct a path to the file	"ConstructorTest.txt"			
3	Read the file content to a string	Path to "ConstructorTest.txt"	N/A	N/A	
	Compare file results with	"ConstructorTest.txt" file	"" (empty string)	"" (empty string)	
4	expected results	content			

Project Name: Project 1: Voting System

Test Stage: Unit _x_ System	Test Date: March 22nd
Test Case ID#: AL2 Test Description:	Name(s) of Testers: Khalid Qasim
This test verifies if the add_line method correctly adds a line to the log	
	Filename: AuditLogUnitTest.cpp
	Testname: AuditLogTest, AddLineTest
	Functions: add line()

Automated: yes X no

Results: Pass X Fail

Preconditions for Test: Audit Log object instantiated and test directory exists

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	· · · · · · · -	testLine = "Test line for AddLineTest";	N/A	N/A	
2	Write the log content to a file	Filename = "AddLineTest.txt"	N/A	N/A	
3		testDirectory and Filename = "AddLineTest.txt"	N/A	N/A	
4	Read the file content to a string	Path to "AddLineTest.txt"	N/A	N/A	
5	I I	"AddLineTest.txt" file content	"Test line for AddLineTest\n"	"Test line for AddLineTest\n"	

Post condition(s)	for Test:
-------------------	-----------

Project Name: Project 1: Voting System	Team# 13

Test Stage: Unit _x_ System __ Test Date: March 22nd

Test Case ID#: AL3 Name(s) of Testers: Khalid Qasim

Test Description:

This test verifies if the clear_log method correctly clears
the logs content to an empty string

Filename: AuditLogUnitTest.cpp
Testname: AuditLogTest, ClearLogTest
Functions: clear_log()

Automated: yes X no

Results: Pass X Fail

Preconditions for Test: Audit Log object instantiated and test directory exists

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	Call clear_log method with the test line	"Test line for ClearLogTest"	N/A	N/A	
2		Filename = "ClearLogTest.txt"	N/A	N/A	
3		testDirectory and Filename = "ClearLogTest.txt"	N/A	N/A	
4	Read the file content to a string	Path to "ClearLogTest.txt"	N/A	N/A	
5	l ^	"ClearLogTest.txt" file content	"" (no content)	"" (no content)	Verify no content in the file

Post condition(s) for Test:

Automated: yes

no X

Project Name: Project 1: Voting System		
Test Stage: Unit _x_ System	Test Date: March 22nd	
Test Case ID#: AL4 Test Description:	Name(s) of Testers: Khalid Qasim	
This test verifies if the add_allocation_table method correformats and adds the allocation table to the log for a CPI election with a tie, this is a visual test	·	

Filename: main.cpp Testname: NA

Functions: add allocation table()

Resu	lts: Pass X	Fail			
Preco	onditions for Test:				
Step	Test Step	Test	Expected	Actual	
.	Description Run the program from main on 3_person_tie_cpl.csv and add to the AuditLog	Data	Result	Result	Notes
	select 'y' to generate audit file Review the data written to audit file against expected outputs for match		True	True	
		4 57 40 0 4			TD # 4.0
Proj	ject Name: Project	1: Voting Syste	em		Team# 13
Test	J	1: Voting Syste		Test Date: March 22nd Name(s) of Testers: Khalid Oasin	
Test Test	Stage: Unit _x_			Test Date: March 22nd Name(s) of Testers: Khalid Qasin	
Test Test Test This	Stage: Unit _x_ Stage: Unit _x_ Stage: Unit _x_ Stage: Case ID#: AL5 Description: test verifies if the add_nats and adds the allocated.	System _allocation_table i ation table to the l	method correctly		
Test Test Test This	Stage: Unit _x_ Stage: Unit _x_ Stage: Unit _x_ Stage: Case ID#: AL5 Description: test verifies if the add_	System _allocation_table i ation table to the l	method correctly		
Test Test Test This form elect	Stage: Unit _x_ Stage: Unit _x_ Stage: Unit _x_ Stage: Case ID#: AL5 Description: test verifies if the add_nats and adds the allocated.	System _allocation_table i ation table to the le visual test	method correctly	Name(s) of Testers: Khalid Qasin Filename: main.cpp	
Test Test Test This form elect	Stage: Unit _x_ Stage: Unit _x_ Stage: Unit _x_ Stage: Case ID#: AL5 Description: test verifies if the add ats and adds the allocation with ties, this is a verified with the stage of the	System _allocation_table i ation table to the le visual test	method correctly	Name(s) of Testers: Khalid Qasin Filename: main.cpp Testname: NA	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
	Run the program from main on				
	sys_test2_opl.csv and add to the AuditLog				
2	select 'y' to generate audit file				
	Review the data written to		True	True	
	audit file against expected				
3	outputs for match				
4					

Project Name: Project 1: Voting System	Team# 13
Test Stage: Unit _x_ System	Test Date: March 22nd
Test Case ID#: AL6 Test Description:	Name(s) of Testers: Khalid Qasim
This test verifies if the tie breaker data is properly sent to the audit log for a CPL election, this is a visual test	
	Filename: main.cpp
	Testname: NA
	Functions: add line(), calculate seats per party()
Automated: yes no X	
Results: Pass X Fail	
Preconditions for Test:	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
	Run the program from main on				
1	3_person_tie_cpl.csv and add				
1	to the AuditLog				
2	select 'y' to generate audit file				
	Review the data written to		True	True	
3	audit file against expected				

	outputs for match		
4			

Project Name: Project 1: Voting System	Team# 13
Test Stage: Unit _x_ System	Test Date: March 22nd
Test Case ID#: AL7 Test Description:	Name(s) of Testers: Khalid Qasim
This test verifies if the tie breaker data is properly sent to the audit log for a OPL election, this is a visual test	
	Filename: main.cpp
	Testname: NA
	Functions: add line(), calculate seats per party()
Automated: yes no_X	
Results: Pass Fail X	
Preconditions for Test:	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	Run the program from main on sys_test2_opl.csv and add to the AuditLog				
2	select 'y' to generate audit file				
	Review the data written to audit file against expected outputs for match		True		See buglist, we are currently unable to send our opl tie-breaker data to the auditlog
4					

Project Name: Project 1: Voting System	Team# 13
Test Stage: Unit _x_ System	Test Date: March 22nd
Test Case ID#: AL8 Test Description: This test verifies if the write_to_file method correctly write the current log to a specified file	Name(s) of Testers: Khalid Qasim
	Filename: AuditLogUnitTest.cpp Testname: AuditLogTest, WriteToFileTest Functions: write_to_file()
Automated: yes X no	
Results: Pass X Fail	

Preconditions for Test: Audit Log object instantiated and test directory exists

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	_	testLine = "Test line for WriteToFileTest";	N/A	N/A	
2		Filename = "WriteToFileTest.txt"	N/A	N/A	
3		testDirectory and Filename = "WriteToFileTest.txt"	N/A	N/A	
4	Read the file content to a string	Path to "WriteToFileTest.txt"	N/A	N/A	
	Compare file results with	"WriteToFileTest.txt" file content	testLine + "\n"	testLine + "\n"	

Post condition(s) for Test:

Project Name: Voting System Team #13

Test Stage:	Unit	_X_	System	Test Date: 3/26/24
-------------	------	-----	--------	---------------------------

Test Case ID#: OPL_ED_Display1 Name(s) of Testers: Connell Hagen

Test Description:

Tests the display() function

setup with:

ElectionData* std case 1 =

ElectionDataParser::create_election("testing/test_data/2_party_opl.csv"); std case 1->display();

Filename: OPLElectionDataUnitTest.cpp **Testname:** OPLElectionDataUnitTest, Display

Functions: display()

Automated: yes X no

Results: Pass X Fail

Preconditions for Test: The ElectionDataParser create_election() function properly sets up an OPL election according to all of its pre- and post-conditions. The winners of the election were properly calculated, and set within the Candidate objects.

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	01.1	testing/test_data/2_party_opl. csv	std_case_1 != nullptr	std_case_1 != nullptr	
2	Display Election results	terminal output	6 Republican: > 4 Alawa - WINNER 2 Etta 3 Democrat: > 2 Pike - WINNER 1 Lucy 0 Beiye	> 4 ALAWA - WINNER 2 Etta	

'display()' outputs all 'to_string()' representations of its aggregated 'Party's to the terminal. All 'Candidate's within the same party are also sorted by number of votes received.

Project Name: Voting System Team #13 Test Stage: Unit X **Test Date: 3/26/24** System Test Case ID#: OPL ED Display2 Name(s) of Testers: Connell Hagen **Test Description:** Tests the display() function's performance setup with: clock tt = clock(); ElectionData* opl 100000 = ElectionDataParser::create election("testing/test data/100000_votes_opl.csv"); opl 100000->display(); const double work time = (clock() - t) / double(CLOCKS PER SEC); Filename: OPLElectionDataUnitTest.cpp **Testname:** OPLElectionDataUnitTest, Display Functions: display()

Preconditions for Test: The ElectionDataParser create_election() function properly sets up an OPL election according to all of its pre- and post-conditions. The winners of the election were properly calculated, and set within the Candidate objects.

Automated: yes X

Results: Pass

no

Fail

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1	Start Timer				
			opl_100000 != nullptr	, — ,	The data will be corrupt if an election is created before this current election is, as it is in
2		testing/test_data/100000_vot es_opl.csv			the test file. This is a bug currently in the buglist.

3	Display Election results	terminal output	69997 Republican:	69997 Republican: > 49995 Alawa - WINNER 20002 Etta 29994 Democrat: > 19993 Pike - WINNER 10001 Lucy 0 Beiye	
4	Check Timer	work_time	work_time <= 4 * 60	work_time <= 4 * 60	

'display()' outputs all 'to_string()' representations of its aggregated 'Party's to the terminal. A 100,000 vote election can be completely tabulated within 4 minutes.

Project Name: Voting System

Test Stage: Unit X

Test Date: 3/26/24

Test Case ID#: OPL_ED_AuditLog1 Name(s) of Testers: Connell Hagen

Test Description:

Tests the generate_audit_log() function

setup with:

ElectionData* test =

ElectionDataParser::create_election("testing/test_data/2_party_opl.csv");

System

Filename: n/a

Testname: OPLElectionDataUnitTest, AuditLog

Functions: generate_audit_file()

Team #13

Automated: yes no X

Results: Pass X Fail

Preconditions for Test: The ElectionDataParser create_election() function properly sets up an OPL election according to all of its pre- and post-conditions. The winners of the election were properly calculated, and set within the Candidate objects.

Step #	Test Step Description	Test Data	±	Actual Result	Notes
1 1	0.1.1	testing/test_data/2_party_opl. csv	test != nullptr	test != nullptr	
2	Generate Audit File		Attention types over total widers 9 bears by for Election 2 works per four-antived Sects 5 perty, were, first Allocation Sects, Remaining Water, Second Allocation, work Nifest 5 memories 3, 6, 1, 3, 1, 200 GM magnification 3, 1, 1, 2, 200 GM magnification 3, 1, 1, 2, 200 GM magnification 3, 1, 1, 2, 200 GM magnification 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,	Election Type: OPL Intal Wotes: 9 Seast by for Election: 2 Seast by for Election: 2 Wotes per Guaranteed Seat: 5 Party, Votes, First Allocation Seats, Remaining Wotes, Second Allocation, Vote %/Seat % Democrat, 3, 0, 3, 1, 30%/SOX 6 Republican: > A Alama - WINMER 2 Etta 1 Democrat: > 2 Pine - WINMER 1 Lucy 0 Bodje 0 Bodje 0 Bodje	

'generate_audit_file()' outputs a header including the type of election and basic statistics relating to it, all ties that were broken during the calculations, a table with a breakdown of the calculation of seat awarding, and all 'to_string()' representations of its aggregated 'Party's to the terminal.

Project Name: Voting System

Test Stage: Unit X System Test Date: 3/26/24

Test Case ID#: OPL_ED_AuditLog2 Name(s) of Testers: Connell Hagen

Test Description:

Tests the generate_audit_log() function

setup with:

ElectionData* test =

ElectionDataParser::create_election("testing/test_data/100000_opl.csv");

Filename: n/a

Testname: OPLElectionDataUnitTest, AuditLog

Functions: generate_audit_file()

Team #13

Automated: yes no X

Results: Pass X Fail

Preconditions for Test: The ElectionDataParser create_election() function properly sets up an OPL election according to all of its pre- and post-conditions. The winners of the election were properly calculated, and set within the Candidate objects.

Step #	Test Step Description	Test Data	_ <u> </u>	Actual Result	Notes
1 1	01.	testing/test_data/100000_opl.	test != nullptr	test != nullptr	
2	Generate Audit File	file output	Direction types of the control was a second allocation, were those the control was second allocation, were those the control was a second allocation, and control was a second allocation and control was a second allocation, were those a second allocation, were those as second allocation and control was a second allocation and control was a second allocation, were those as second allocation and control was a second allocation, were those as second allocation and control was a second allocatio	Election Type: GPT total Votes: 100000 Seafs by for Election: 2 Votes per Guarnteed Seaf; 50000 Party, Votes, First Allocation Seafs, Remaining Wotes, Second Allocation, Vote %/Seaf % Democrat, 20094, 0, 20094, 1, 20%/SGK SUPPLICE, 20097, 1, 19097, 0, 60%/SGK SUPPLICE, 20097, 1, 19097, 1, 19097, 0, 60%/SGK SUPPLICE, 20097, 1, 19097, 1, 19097, 0, 60%/SGK SUPPLICE, 2009, 1000,	

'generate_audit_file()' outputs a header including the type of election and basic statistics relating to it, all ties that were broken during the calculations, a table with a breakdown of the calculation of seat awarding, and all 'to_string()' representations of its aggregated 'Party's to the terminal.

Project Name: Voting System

Team #13

Test Stage: Unit _X_ System __ Test Date: 3/26/24

Test Case ID#: CPL_ED_Display1 Name(s) of Testers: Connell Hagen

Test Description:

Tests the display() function

setup with:

ElectionData* cpl 1p =

ElectionDataParser::create_election("testing/test_data/1_person_cpl.csv");

cpl 1p->display();

Filename: CPLElectionDataUnitTest.cpp

Testname: CPLElectionDataUnitTest, Display

Functions: display()

Automated: yes_X_ no

Results: Pass X Fail

Preconditions for Test: The ElectionDataParser create_election() function properly sets up a CPL election according to all of its pre- and post-conditions. The winners of the election were properly calculated, and set within the Candidate objects.

Step	Test Step	Test	Expected	Actual	Notes
#	Description	Data	Result	Result	
1	Create CPLElectionData Object	testing/test_data/1_person_cp l.csv	cpl_1p != nullptr	cpl_1p != nullptr	

			2 Democratic:	2 Democratic:	
2	Display Election results	terminal output	> Gary - WINNER	> Gary - WINNER	

'display()' outputs all 'to_string()' representations of its aggregated 'Party's to the terminal.

Project Name:	Voting System	

System

Test Date: 3/26/24

Test Case ID#: CPL_ED_Display2

Name(s) of Testers: Connell Hagen

Team #13

Test Description:

Tests the display() function

Test Stage: Unit X

setup with:

ElectionData* std case 1 =

ElectionDataParser::create_election("testing/test_data/sys_test3_cpl.csv");

std_case_1->display();

Filename: CPLElectionDataUnitTest.cpp

Testname: CPLElectionDataUnitTest, Display

Functions: display()

Automated: yes X no

Results: Pass X Fail

Preconditions for Test: The ElectionDataParser create_election() function properly sets up a CPL election according to all of its preand post-conditions. The winners of the election were properly calculated, and set within the Candidate objects.

Step #	Test Step Description	Test Data	1	Actual Result	Notes
1	Create CPLElectionData Object	testing/test_data/sys_test3_cp l.csv	std_case_1 != nullptr	std_case_1 != nullptr	

_		i	i	
			3 Democratic:	3 Democratic:
			> Joe - WINNER	> Joe - WINNER
			Sally	Sally
			Ahmed	Ahmed
			2 Republican:	2 Republican:
			> Allen - WINNER	> Allen - WINNER
			Nikki	Nikki
			Taihui	Taihui
			2 Reform:	2 Reform:
			> Xinyue - WINNER	> Xinyue - WINNER
			Nikita	Nikita
			1 Green:	1 Green:
			Bethany	Bethany
			1 Independent:	1 Independent:
			Mike	Mike
			0 New Wave:	0 New Wave:
2	Display Election results	terminal output	Sarah	Sarah

`display()` outputs all `to_string()` representations of its aggregated `Party`s to the terminal.

Project Name: Voting System	Team #13
Test Stage: Unit _X_ System	Test Date: 3/26/24
Test Case ID#: CPL_ED_Display3 Test Description: Tests the display() function's performance	Name(s) of Testers: Connell Hagen
setup with: clock_t t = clock(); ElectionData* cpl_100000 = ElectionDataParser::create_election("testing/test_data/100000_votes_cpl.csv"); cpl_100000->display(); const double work_time = (clock() - t) / double(CLOCKS_PER_SEC);	
Automated: yes_X_ no	Filename: CPLElectionDataUnitTest.cpp Testname: CPLElectionDataUnitTest, Display Functions: display()
Results: Pass X Fail	
Preconditions for Test: The ElectionDataParser create_election() function proper and post-conditions. The winners of the election were properly calculated, and set	, ,

Step #	Test Step Description	Test Data	1	Actual Result	Notes
1	Start Timer				
2	~	testing/test_data/100000_vot es_cpl.csv	cpl_100000 != nullptr	cpl_100000 != nullptr	

			30257 Republican:	30257 Republican:	
			> Allen - WINNER	> Allen - WINNER	
			Nikki	Nikki	
			Taihui	Taihui	
			29664 Democratic:	29664 Democratic:	
			> Joe - WINNER	> Joe - WINNER	
			Sally	Sally	
			Ahmed	Ahmed	
			20022 Reform:	20022 Reform:	
			> Xinyue - WINNER	> Xinyue - WINNER	
			Nikita	Nikita	
			10516 Independent:	10516 Independent:	
			Mike	Mike	
			10029 Green:	10029 Green:	
			Bethany	Bethany	
			0 New Wave:	0 New Wave:	
	Di I El ci la		Sarah	Sarah	
3	Display Election results	terminal output			
4	Check Timer	work time	work_time <= 4 * 60	work_time <= 4 * 60	

'display()' outputs all 'to_string()' representations of its aggregated 'Party's to the terminal. A 100,000 vote election can be completely tabulated within 4 minutes.

Project Name:	Voting System	Team #13
.7	₽ •/	

Test Stage: Unit _X_ System __ Test Date: 3/26/24

Test Case ID#: CPL_ED_AuditLog1 Name(s) of Testers: Connell Hagen

Test Description:

Tests the generate_audit_log() function

setup with:

ElectionData* test =

ElectionDataParser::create_election("testing/test_data/1_person_cpl.csv");

Filename: n/a

Testname: CPLElectionDataUnitTest, AuditLog

Functions: generate audit file()

Automated: yes no X

Results: Pass X Fail

Preconditions for Test: The ElectionDataParser create_election() function properly sets up a CPL election according to all of its preand post-conditions. The winners of the election were properly calculated, and set within the Candidate objects.

Step #	Test Step Description	Test Data	1	Actual Result	Notes
1	Create CPLElectionData	testing/test_data/1_person_cp	test != nullptr	test != nullptr	
1	Object	l.csv	Haction types of the test of t	Election Type: CPL Local Words: 2 Lo	
2	Generate Audit File	file output		2 Democratic: > Gary - WINNER	

'generate_audit_file()' outputs a header including the type of election and basic statistics relating to it, all ties that were broken during the calculations, a table with a breakdown of the calculation of seat awarding, and all 'to_string()' representations of its aggregated 'Party's to the terminal.

Project Name: Voting System

Team #13

Test Stage: Unit _X_ System __

Test Date: 3/26/24

Test Case ID#: CPL_ED_AuditLog2

Name(s) of Testers: Connell Hagen

Test Description:

Tests the generate_audit_log() function

setup with:

ElectionData* test =

ElectionDataParser::create_election("testing/test_data/sys_test3_cpl.csv");

Filename: n/a

Testname: CPLElectionDataUnitTest, AuditLog

Functions: generate audit file()

Automated: yes no X

Results: Pass X Fail

Preconditions for Test: The ElectionDataParser create_election() function properly sets up a CPL election according to all of its preand post-conditions. The winners of the election were properly calculated, and set within the Candidate objects.

Step #	Test Step Description	Test Data	r	Actual Result	Notes
1	Create CPLElectionData Object	testing/test_data/sys_test3_cp l.csv	test != nullptr	test != nullptr	

			Hiertion types CPL Table to for Election: 3 Works pro General Control Works pro General Control Forey, Works, First Allocation south, Remaining Works, Second Allocation, Work Notest X GROUND CONTROL Forey, Works, First Allocation south, Remaining Works, Second Allocation, Work Notest X Forey, Works, First Allocation south, Remaining Works, Second Allocation, Work Notest Forey, Works, First Allocation south, Remaining Works, Second Allocation, Work Notest Forey, Works, First Allocation Forey, Wor	nearly, votes, first allocation seats, Remaining Votes, Second Allocation, vote %/Seat % Democratic, 3, 10, 0, 32%/JCM Creen, 1, 0, 1, 0, 110/KB Creen, 1, 0, 1, 0, 110/KB Creen, 1, 0, 1, 0, 110/KB Creen, 1, 0, 1, 0, 10, 110/KB Creen, 2, 0, 2, 0, 2, 10, 22%/JSK Republican, 2, 0, 2, 1, 22%/JSK Republican, 2, 10, 2, 1, 2,	
2	Generate Audit File	file output			

'generate_audit_file()' outputs a header including the type of election and basic statistics relating to it, all ties that were broken during the calculations, a table with a breakdown of the calculation of seat awarding, and all 'to_string()' representations of its aggregated 'Party's to the terminal.

Project Name: Voting System

Test Stage: Unit _X_ System__ Test Date: 3/26/24

Test Case ID#: CPL_ED_AuditLog3 Name(s) of Testers: Connell Hagen

Tests the generate_audit_log() function

setup with:
ElectionData* test =
ElectionDataParser::create_election("testing/test_data/100000_votes_cpl.csv.csv");

Filename: n/a
Testname: CPLElectionDataUnitTest, AuditLog
Functions: generate audit file()

Preconditions for Test: The ElectionDataParser create_election() function properly sets up a CPL election according to all of its preand post-conditions. The winners of the election were properly calculated, and set within the Candidate objects.

Automated: yes

Results: Pass

no X

Fail

Step #	Test Step Description	Test Data	1	Actual Result	Notes
1	01.	testing/test_data/100000_vot es_cpl.csv.csv	test != nullptr	test != nullptr	

2	Generate Audit File		Direction types: CAN COLD VARIATION TO LINE TO THE COLD VARIATION TO LINE TO THE COLD VARIATION TO LINE TO THE COLD VARIATION TO LINE	Election Type: CR total locks: 180000 Scalts by for Election: 3 Party, votes, First Allocation Scats, Remaining Votes, Second Allocation, Vote %/Seat % Party, votes, First Allocation Scats, Remaining Votes, Second Allocation, Vote %/Seat % Democratic, 29064, 0, 29064, 1, 290/31% Foreign Scales, 1, 1907, 0, 1	
---	---------------------	--	---	--	--

'generate_audit_file()' outputs a header including the type of election and basic statistics relating to it, all ties that were broken during the calculations, a table with a breakdown of the calculation of seat awarding, and all 'to_string()' representations of its aggregated 'Party's to the terminal.

Project Name: Project 1: Voting System Team#13

Test Stage: Unit __ System X Test Date: 22/03/2024

Name(s) of Testers: Michael Mulhall, Grant Oie, Connell

Test Case ID#: S 1 Hagan, and Khalid Qasim

Test Description: Ballot file with 1 person for a CPL election.

repo-Team13/Project1/testing/testing_data/1_person_cpl.csv

Automated: yes X no

Results: Pass X Fail

Preconditions for Test: None

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election			
		_Data e =			
1 2	Election is created with correct	Create_Election("1_person_c			
	file name	pl.csv'')			
	Election results are displayed		CPL Display test passes	CPL Display test passes	
3		e.display()			
4					
5					

Post condition(s) for Test:

Project Name: Project 1: Voting System Team#13

Test Stage: Unit __ System X Test Date: 22/03/2024

Name(s) of Testers: Michael Mulhall, Grant Oie, Connell

Test Case ID#: S 2 Hagan, and Khalid Qasim

Test Description: Ballot file with 1 person for a OPL election.

Automated: yes X no	repo-Team13/Project1/testing/testing_data/1_person_opl.csv
Results: Pass X Fail	
Preconditions for Test: None	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election			
		_Data e =			
1 2	Election is created with correct	Create_Election("1_person_o			
	file name	pl.csv'')			
	Election results are displayed		OPL Display test passes	OPL Display test passes	
3	into the terminal	e.display()		î î î	
4					
5					

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit System X	Test Date: 22/03/2024 Name(s) of Testers: Michael Mulhall, Grant Oie, Connell
Test Case ID#: S 3	Hagan, and Khalid Qasim
Test Description: Ballot file with 2 parties and multiple candidates for an OPL Election	
Automated: yes X no	repo-Team13/Project1/testing/testing_data/2_party_opl.csv
Results: Pass X Fail	
Preconditions for Test: None	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election Data e =			
	Election is created with correct file name				
	Election results are displayed into the terminal	e.display()	OPL Display test passes	OPL Display test passes	
4					
5					

Project Name: Project 1: Voting Sys	em Team#13
-------------------------------------	------------

Test Stage: Unit __ System X Test Date: 22/03/2024

Name(s) of Testers: Michael Mulhall, Grant Oie, Connell

Test Case ID#: S 4 Hagan, and Khalid Qasim

Test Description: Ballot file with 2 people, one in each party,

and they tied.

repo-Team13/Project1/testing/testing data/2 people cpl tie.cs

V

Automated: yes X no

Results: Pass X Fail

Preconditions for Test: None

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					

			Election			
			_Data e =			
	_	Election is created with correct	Create_Election("2_people_c			
			pl tie.csv")			
		Election results are displayed		CPL Display test passes	CPL Display test passes	
	3	into the terminal	e.display()			
	4					
5						

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit System X	Test Date: 22/03/2024 Name(s) of Testaves, Michael Mulhall, Grant Oic Connell
Test Case ID#: S 5	Name(s) of Testers: Michael Mulhall, Grant Oie, Connell Hagan, and Khalid Qasim
Test Description: Ballot file with 3 people, all in different parties, and they tied	
	repo-Team13/Project1/testing/testing_data/3_person_tie_cpl.cs
Automated: yes X no	V
Results: Pass X Fail	
Preconditions for Test: None	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election			
		Data e =			
	Election is created with correct	Create_Election("3_person_t			
	file name	ie cpl.csv")			
	Election results are displayed		CPL Display test passes	CPL Display test passes	
3	into the terminal	e.display()	- · · · ·		
4					

5			

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit System X Test Case ID#: S 6 Test Description: Ballot file with 100,000 votes for a CPL Election.	Test Date: 22/03/2024 Name(s) of Testers: Michael Mulhall, Grant Oie, Connell Hagan, and Khalid Qasim
Automated: yes X no	repo-Team13/Project1/testing/testing_data/100000_votes_cpl.
Results: Pass X Fail	
Preconditions for Test: None	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election Data e =			
2	Election is created with correct file name	Create_Election("100000_vo tes cpl.csv")			
	Election results are displayed into the terminal	e.display()	CPL Display test passes	CPL Display test passes	
4					
5					

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit System X	Test Date: 22/03/2024 Name(s) of Testers: Michael Mulhall, Grant Oie, Connell
Test Case ID#: S 7 Test Description: Ballot file with 100,000 votes for a OPL Election.	Hagan, and Khalid Qasim
	repo-Team13/Project1/testing/testing_data/100000_votes_opl.c
Automated: yes X no	
Results: Pass X Fail	
Preconditions for Test: None	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election Data e =			
2	Election is created with correct file name	Create_Election("100000_votes_opl.csv")			
3	Election results are displayed into the terminal	e.display()	OPL Display test passes	OPL Display test passes	
4					
5					

Project Name: Project 1: Voting System Team#13

Test Stage: Unit __ System X Test Date: 22/03/2024

Name(s) of Testers: Michael Mulhall, Grant Oie, Connell

Test Case ID#: S 8 Hagan, and Khalid Qasim

Test Description: Ballot file with 3 parties, 6 candidates, and no ties	
Automated: yes X no	repo-Team13/Project1/testing/testing_data/sys_test1_opl.csv
Results: Pass X Fail	
Preconditions for Test: None	

Step # 1	Test Step Description	Test Data	Expected Result	Actual Result	Notes
	Election is created with correct	Election _Data e = Create_Election("sys_test1_o pl.csv")			
	Election results are displayed into the terminal	e.display()	OPL Display test passes	OPL Display test passes	
4					
5					

Project Name:	Project 1:	Voting System	Team#13
----------------------	------------	----------------------	---------

Test Stage: Unit __ System X Test Date: 22/03/2024

Name(s) of Testers: Michael Mulhall, Grant Oie, Connell

Test Case ID#: S 9 Hagan, and Khalid Qasim

Test Description: Ballot file with 3 parties, 8 candidates, and

no ties

repo-Team13/Project1/testing/testing_data/sys_test2_opl.csv

Automated: yes X no

Results: Pass X	Fail		
Preconditions for T	est: None		

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election Data e =			
	Election is created with correct file name	Create_Election("sys_test2_o pl.csv")			
	Election results are displayed into the terminal	e.display()	OPL Display test passes	OPL Display test passes	
4					
5					

Project Name: Project 1: Voting System	Team#13
---	---------

Test Stage: Unit System X Test Date: 22/03/2024

Name(s) of Testers: Michael Mulhall, Grant Oie, Connell

Test Case ID#: S 10 Hagan, and Khalid Qasim

Test Description: Ballot file with 6 parties, 9 votes, and a

quota of 3.

repo-Team13/Project1/testing/testing_data/sys_test3_cpl.csv

Automated: yes X no

Results: Pass X Fail

Preconditions for Test: None

Step	Test Step	Test	Expected	Actual	
------	-----------	------	----------	--------	--

#	Description	Data	Result	Result	Notes
1					
		Election			
	Election is created with correct file name	_Data e = Create_Election("sys_test3_c pl.csv")			
	Election results are displayed		CPL Display test passes	CPL Display test passes	
3	into the terminal	e.display()			
4					
5					

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit System X Test Case ID#: S 11 Test Description: Ballot file with 4 candidates, 4 parties, and no ties	Test Date: 22/03/2024 Name(s) of Testers: Michael Mulhall, Grant Oie, Connell Hagan, and Khalid Qasim
Automated: yes X no	repo-Team13/Project1/testing/testing_data/sys_test4_cpl.csv
Results: Pass X Fail	
Preconditions for Test: None	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election			
1		_Data e =			
	Election is created with correct	Create_Election("sys_test4_c			
	file name	pl.csv")			
	Election results are displayed		CPL Display test passes	CPL Display test passes	
3	into the terminal	e.display()	- · · · ·		

4			
5			

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit System X	Test Date: 22/03/2024 Name(a) of Testava, Michael Mulhell Guant Oic Connell
Test Case ID#: S 12	Name(s) of Testers: Michael Mulhall, Grant Oie, Connell Hagan, and Khalid Qasim
Test Description: Ballot file with a quota of 3, 2 parties, 13 votes, and all but 1 vote go to one party.	
	repo-Team13/Project1/testing/testing_data/sys_test5_overload cpl.csv
Automated: yes X no	
Results: Pass X Fail	
Preconditions for Test: None	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election			
1		_Data e =			
	Election is created with correct	Create_Election("sys_test5_o			
	file name	verload_cpl.csv")			
	Election results are displayed		CPL Display test passes	CPL Display test passes	
		e.display()		1 7 1	
4					
5					

Project Name: Project 1: Voting System	Team#13
Test Stage: Unit System X	Test Date: 22/03/2024 Name(s) of Tostova, Michael Mulhell Crant Oic Connell
Test Case ID#: S 13	Name(s) of Testers: Michael Mulhall, Grant Oie, Connell Hagan, and Khalid Qasim
Test Description: Ballot file with 4 candidates, 4 parties, and tying for OPL	ringun, unu rinunu Quom
	repo-Team13/Project1/testing/testing data/sys test6 opl.csv
Automated: yes X no	reporteum 19/11 ojecti/testing/testing_data/sys_testo_opices/
Results: Pass X Fail	
Preconditions for Test: None	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election			
2	Election is created with correct file name	_Data e = Create_Election("sys_test6_o verloadseats_opl.csv")			
3	Election results are displayed into the terminal	e.display()	OPL Display test passes	OPL Display test passes	
4					
5					

Project Name: Project 1: Voting System Team#13

Test Stage: Unit __ System X Test Date: 22/03/2024

Name(s) of Testers: Michael Mulhall, Grant Oie, Connell

Test Case ID#: S 14 Hagan, and Khalid Qasim

Test Description: Ballot file with a quota of 4, 8 votes, and 3 parties where every vote goes to the same party for CPL	
	repo-Team13/Project1/testing/testing_data/sys_test7_votesforsingleparty_cpl.csv
Automated: yes X no	
Results: Pass X Fail	
Preconditions for Test: None	

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election			
		_Data e =			
	Election is created with correct	Create_Election("sys_test7_v			
	file name	otesforsingleparty cpl.csv")			
	Election results are displayed		CPL Display test passes	CPL Display test passes	
3	into the terminal	e.display()			
4					
5					

Project Name: Project 1: Voting System	Team#13		
Test Stage: Unit System X Test Case ID#: S 15 Test Description: Ballot file with a quota of 4, 8 votes, and 3 parties where every vote goes to the same party for OPL	Test Date: 22/03/2024 Name(s) of Testers: Michael Mulhall, Grant Oie, Connell Hagan, and Khalid Qasim		
Automated: yes X no	repo-Team13/Project1/testing/testing_data/sys_test8_votesfors inglecandidate opl.csv		

Results: Pass X Fail		
Preconditions for Test: None		

Step	Test Step	Test	Expected	Actual	
#	Description	Data	Result	Result	Notes
1					
		Election _Data e = Create_Election("sys_test8_v			
2	Election is created with correct file name	otesforsinglecandidate_opl.cs v")			
3	Election results are displayed into the terminal	e.display()	OPL Display test passes	OPL Display test passes	
4					
5					