Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit 1_ System __ **Test Date: 2021/3/13**

Test Case ID#: IR_Haswinner_1 **Name(s) of Testers: Chenxuan Liu**

Test Description: This test is created to test the functionality of the haswinner() in the file IR_sysTest.java. This function determines whether a winner exists in the election. If yes, return the candidate.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass _1___ Fail_____

Preconditions for Test:
A created ArrayList<Candidate> contains all the candidates in the voting
A created ArrayList<Party> contains all the parties in the voting
Created IR_Ballot objects.
Created Candidate and Party object

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Call assertEqual() function to test the winner.	can2, ir.haswinner()	can2	can2	
3					
4					

Post condition(s) for Test:
The candidate who wins the election is found and nothing changed.

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit 1_ System __ **Test Date: 2021/3/13**

Test Case ID#: IR_get_least_candidate_1 **Name(s) of Testers: Chenxuan Liu**

Test Description: Test the functionality of get_least_candidate() in IRsys class in the file IR_sysTest.java. This function finds the candidate has the least ballot. If find, return the candidate index.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass _1___ Fail_____

Preconditions for Test:
A created ArrayList<Candidate> contains all the candidates in the voting
A created ArrayList<Party> contains all the parties in the voting
Created IR_Ballot objects.
Created Candidate and Party object

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual() to test the result	2, ir.get_leastcandidate()	2	2	
3					
4					

Post condition(s) for Test:
None.

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: Party_getVote_1 **Name(s) of Testers: Chenxuan Liu**

Test Description: This test is created for testing the functionality of the getVote() function in the Party Class in the file PartyTest.java. The goal is to return the votes count for the specific party object

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test: A party object is created.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual() to test new created party object	None	0	0	Passed
3					
4					

Post condition(s) for Test: None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: Party_setVote_1 **Name(s) of Testers: Chenxuan Liu**

Test Description: Test the functionality of the setVote() function in Party class in the file PartyTest.java.. The goal is to update the rank of the party object by the given integer.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created Party object.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	set party object vote to 100	100			
3	Use assertEqual() function to test the function		100	100	
4	set the party object vote to -1	-1			
5	Use assertEqual() function to test the function		-1	-1	For the case in OPL Voting system

Post condition(s) for Test:
The party object vote field will be set to the number in the setVote() function.

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: Party_getName_1 **Name(s) of Testers: Chenxuan Liu**

Test Description: Test the functionality of the getName() function in Party class in the file PartyTest.java. The goal for this is to find the name of the Party object.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created Party object.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual function to test the return string of the function to the expected name		p1	p1	
3					
4					
5					

Post condition(s) for Test:
None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __

Test Date: 2021/3/13

Test Case ID#: Party_addMember_1

Name(s) of Testers: Chenxuan Liu

Test Description: Test the function of addMember() in the party class in the file PartyTest.java. The goal is to add a Candidate object to the field member, which type is ArrayList<Candidate>.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created Party object.
Have a created Candidate object.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual function to test the expected ArrayList and the party field member. Use function getMember() to access the field	can1 can2	[can1,can2]	[can1,can2]	
3					
4					
5					

Post condition(s) for Test:
The member field of the party object is updated with the input candidate object.

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __

Test Date: 2021/3/13

Test Case ID#: Party_getMember_1

Name(s) of Testers: Chenxuan Liu

Test Description: Test the function of getMember() in the party class in the file PartyTest.java. The goal is to get the Candidate ArrayList.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created Party object.
Have a created Candidate object.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Use assertEqual function to test the expected ArrayList and the party field member. Use function getMember() to access the field	can1 can2	[can1,can2]	[can1,can2]	
2					
3					
4					
5					

Post condition(s) for Test: None

Project Name: Project 1: Voting System **Team#11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: IR_redistribution_1 **Name(s) of Testers: Chenxuan Liu**

Test Description: This test is created to test the functionality of the redistribution() function. It is in the file IR_sysTest.java.This function finds the lowest vote candidate and redistributes the vote from the least candidate.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
A created ArrayList<Candidate> contains all the candidates in the voting
A created ArrayList<Party> contains all the parties in the voting
Created IR_Ballot objects.
Created Candidate and Party object

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	call the redistribution function				
3	use assertEqual() function to test the left candidate votes	3 can2.getVote()	3	3	
4	use assertEqual() function to test the left candidate votes	2 can1.getVote()	2	2	

Post condition(s) for Test:

Candidates votes get resigned by redistributing the least candidate vote.

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit 1_ System __ **Test Date: 2021/3/14**

Test Case ID#: OPL_Firstroundseats_1 **Name(s) of Testers: Chenxuan Liu**

Test Description: This test is created for testing the functionality of the firstround_Seats() function in OPL_sys class. The goal for this function is to return an ArrayList of Integer that contains the seats selected by each party.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Created candidate objects
Created party objects
Created party ArrayList that filled with the party objects
Created candidate ArrayList filled with the candidate objects
Every needed field is assigned to a meaningful number

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Created expect vaue				
3	Use assertEqual function to test the difference between the function call and the expected arraylist	expected opl1.firstround_Seats()	[1,1]	[1,1]	
4					

Post condition(s) for Test:

The returned ArrayList contains the first round assigned seats.

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit 1_ System __ **Test Date: 2021/3/14**

Test Case ID#: OPL_Checkremainingseats_1 **Name(s) of Testers: Chenxuan Liu**

Test Description: This test is created for testing the functionality of the checkRemainSeats() function in OPL_sys class. The goal for this function is to return a boolean after the firstround_seat() function call to determine if any remain seats left.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Created candidate objects
Created party objects
Created party ArrayList that filled with the party objects
Created candidate ArrayList filled with the candidate objects
Every needed field is assigned to a meaningful number

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Created expect vaue				
3	Use assertEqual function to test the difference between the function call and the expected arraylist	expected opl1.firstround_Seats()	[1,1]	[1,1]	
4	Creat a new opl2 object and add a new Candidate object can4				
5	call firstround_Seats() function to this new opl2 object				
6	Use assertTrue() function to checkt the boolean of	opl2.checkRemainSeats()	True	True	

	checkRemainSeeats()				

Post condition(s) for Test:

None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit 1_ System __ **Test Date: 2021/3/14**

Test Case ID#: OPL_findlargestvote_1 **Name(s) of Testers: Chenxuan Liu**

Test Description: This test is created for testing the functionality of the findlargestvote() function in OPL_sys class. The goal for this function is to return an index refer to the party that receives the most remaining votes.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Created candidate objects
Created party objects
Created party ArrayList that filled with the party objects
Created candidate ArrayList filled with the candidate objects
Every needed field is assigned to a meaningful number

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertTrue() function to check the tie condition	opl1.findlargestvote()>=0	True	True	
3	Use assertTrue() function to check the tie condition	opl1.findlargestvote() <2	True	True	
4	creat a new opl2 object				
5	use assertEqual() function to check the function call and the expected value	1, opl2.findlargestvote()	1	1	

Post condition(s) for Test:

The returned int is the party index has the most vote.

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit 1_ System __ **Test Date: 2021/3/14**

Test Case ID#: OPL_secondroundseats_1 **Name(s) of Testers: Chenxuan Liu**

Test Description: This test is created for testing the functionality of the secondround_seats() function in OPL_sys class. The goal for this function is to return an ArrayList of Integer that contains the seats selected by each party after the second round seat assignment.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Created candidate objects
Created party objects
Created party ArrayList that filled with the party objects
Created candidate ArrayList filled with the candidate objects
Every needed field is assigned to a meaningful number

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Creat a new opl2 objects				
3	call firstround_Seats() to the opl2 object				
4	use assertEqual() function to test the expected value and function call	expect opl2.secondround_seats(input) input = opl2.firstround_Seats()	[1,2]	[1,2]	

Post condition(s) for Test:

The returned ArrayList contains the second round assigned seats.

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit 1_ System __ **Test Date: 2021/3/14**

Test Case ID#: OPL_Findwinner_1 **Name(s) of Testers: Chenxuan Liu**

Test Description: This test is created for testing the functionality of the findwinner() function in OPL_sys class. The goal for this function is to return an ArrayList of Candidate that contains the selected winner.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Created candidate objects
Created party objects
Created party ArrayList that filled with the party objects
Created candidate ArrayList filled with the candidate objects
Every needed field is assigned to a meaningful number

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	use assertEqual for simple case	expect opl1.findwinner(first) first = opl1.firstround_Seats	[can1,can3]	[can1, can3]	
3	Creat new opl2 objects				
4	Use assertEqual for a complex case	expect opl2.findwinner(sceond) second = opl2.secondround_seats(first) first = opl2.firstround_Seats()	[can1, can3, can4]	[can1, can3, can4]	

Post condition(s) for Test:
The returned ArrayList contains the selected winners.
.

Project Name: Project 1: Voting System **Team#11**

Test Stage: Unit __ System _1_ **Test Date: 03/14/2021**

Test Case ID#: IR_direct_winner **Name(s) of Testers: Zilong He**

Test Description:

manually run the voting system with IR_direct_winner.csv

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes___ no _1__

Results: Pass __1___ Fail_____

Preconditions for Test:
A CSV file includes the all the election information

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Run Voting_System.java				
3	Input the file name in the console	"IR_direct_winner" in the CSV folder.	Carroll(R) wins the election	The winner is Carroll from R	An audit file with all the ballot distribution recorded
4					

Post condition(s) for Test:

Project Name: Project 1: Voting System **Team#11**

Test Stage: Unit __ System _1_ **Test Date: 03/14/2021**

Test Case ID#: IR_popularity **Name(s) of Testers: Zilong He**

Test Description:
manually run the voting system with IR_popularity.csv

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes___ no _1__

Results: Pass __1___ Fail_____

Preconditions for Test:
A CSV file includes the all the election information

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Run Voting_System.java				
3	Input the file name in the console	"IR_popularity" in the CSV folder.	Gaskell(R) wins the election	The winner is Gaskell from R	An audit file with all the ballot distribution recorded
4					

Post condition(s) for Test:

Project Name: Project 1: Voting System **Team#11**

Test Stage: Unit __ System _1_ **Test Date: 03/14/2021**

Test Case ID#: IR_worstcase_tie **Name(s) of Testers: Zilong He**

Test Description:

manually run the voting system with IR_worsecase_tie.csv

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes___ no _1__

Results: Pass ___1__ Fail_____

Preconditions for Test:
A CSV file includes the all the election information

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Run Voting_System.java				
3	Input the file name in the console	"IR_worstcase_tie" in the CSV folder.	Ben(D) or Gaskell(R) or Madge(R) wins the election	The winner is Gaskell from R	An audit file with all the ballot distribution recorded
4					

Post condition(s) for Test:

Project Name: Project 1: Voting System **Team#11**

Test Stage: Unit __ System _1_

Test Date: 03/14/2021

Test Case ID#: OPL_normal_candidate-tie

Name(s) of Testers: Zilong He

Test Description:
manually run the voting system with OPL_normal_candidate-tie.csv

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes___ no _1__

Results: Pass __1___ Fail_____

Preconditions for Test:
A CSV file includes all the election information

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Run Voting_System.java				
3	Input the file name in the console	"OPL_normal_candidate-tie" in the CSV folder.	First round D:1, R:1, I:1. Second round D+1. Abraham from the party D, Bach from the party D Holmes from the party R, Walkley from the party I get seats	Abraham from the party D Bach from the party D Holmes from the party R Walkley from the party I	An audit file with all the ballot distribution recorded
4					

Post condition(s) for Test:

Project Name: Project 1: Voting System **Team#11**

Test Stage: Unit __ System _1_

Test Date: 03/14/2021

Test Case ID#: OPL_overseats_doubleties

Name(s) of Testers: Zilong He

Test Description:

manually run the voting system with OPL_overseats_doubleties.csv

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes___ no __1_

Results: Pass __1___ Fail_____

Preconditions for Test:
A CSV file includes all the election information

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Run Voting_System.java				
3	Input the file name in the console	"OPL_overseats_doubleties" in the CSV folder.	First round D:1, R:1, I:2 (Over seats). Second round D and R tie, coin flip to determine to get a seat. Second round D:2, R:2, I:1.	Pike from the party D Foster from the party D Wesley from the party R Landon from the party R Edmund from the party I	An audit file with all the ballot distribution recorded
4					

Post condition(s) for Test:

Project Name: Project 1: Voting System **Team#11**

Test Stage: Unit __ System _1_ **Test Date: 03/14/2021**

Test Case ID#: OPL_shortcase **Name(s) of Testers: Zilong He**

Test Description:

manually run the voting system with OPL_shortcase.csv

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes___ no _1__

Results: Pass __1___ Fail_____

Preconditions for Test:
A CSV file includes all the election information

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Run Voting_System.java				
3	Input the file name in the console	"OPL_shortcase" in the CSV folder.	First round not seat allocated.. Second R gets 1 seat and allocated to Madge	Madge from the party R	An audit file with all the ballot distribution recorded
4					

Post condition(s) for Test:

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: Candidate_getVote_1 **Name(s) of Testers: Yingwen Weng**

Test Description: This test is created for testing the functionality of the getVote() function in the Candidate Class. The goal is to return the votes count for the specific candidate object

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test: A candidate object is created.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual() to test new created Candidate object	None	0	0	Passed
3					
4					

Post condition(s) for Test: None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: Candidate_setVote_1 **Name(s) of Testers: Yingwen Weng**

Test Description: Test the functionality of the setVote() function in Candidate class. The goal is to update the rank of the candidate object by the given integer.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created candidate object.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	set candidate object vote to 1	1			
3	Use assertEqual() function to test the function		1	1	
4	set the party object vote to -1	-1			
5	Use assertEqual() function to test the function		-1	-1	
6	set the party object vote to 0	0			
7	Use assertEqual() function to test the function		0	0	

Post condition(s) for Test:
The candidate object vote field will be set to the number in the setVote() function.

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: Candidate_getName_1 **Name(s) of Testers: Yingwen Weng**

Test Description: Test the functionality of the getName() function in Candidate class. The goal for this is to find the name of the candidate object.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created candidate object.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual function to test the return string of the function to the expected name		Jack	Jack	
3	Use assertEqual function to test the return string of the function to the expected name		Jerry	Jerry	
4					
5					

Post condition(s) for Test:
None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: Candidate_getParty_1 **Name(s) of Testers: Yingwen Weng**

Test Description: Test the functionality of the getParty() function in Candidate class. The goal for this is to get the party's name of the candidate object.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created candidate object.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual function to test the return string of the function to the expected party's name		p1	p1	
3					
4					
5					

Post condition(s) for Test:
None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: Candidate_getballots_1 **Name(s) of Testers: Yingwen Weng**

Test Description: Test the functionality of the getballots() function in Candidate class. The goal for this is to get the ballots of the candidate object.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created candidate object.
Have an IR_Ballot ArrayList with 1 ballot in it.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual function to test the expected ArrayList and the candidate field member. Use function getballots() to access the field		An IR_Ballot ArrayList with 1 ballot in it.	An IR_Ballot ArrayList with 1 ballot in it.	
3					
4					
5					

Post condition(s) for Test:
None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: Candidate_addIRballot_1 **Name(s) of Testers: Yingwen Weng**

Test Description: Test the functionality of the addIRballot() function in Candidate class. The goal for this is to add IR_ballots to candidate object.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created candidate object.
Have an IR_Ballot ArrayList.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual function to test the expected ArrayList and the candidate field member. Use function getballots() to access the field		An IR_Ballot ArrayList with 2 ballots in it.	An IR_Ballot ArrayList with 2 ballots in it.	
3					
4					
5					

Post condition(s) for Test:
None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: Coin_Flip_flip_1 **Name(s) of Testers: Yingwen Weng**

Test Description: Test the functionality of the flip() function in Coin_Flip class. The goal for this is to get the return number of the flip function.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
None.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertTrue function to test the return number of the function to the expected range.		true	true	
3	Use assertTrue function to test the return number of the function to the expected range.		true	true	
4					
5					

Post condition(s) for Test:
None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: IR_Ballot_getRank_1 **Name(s) of Testers: Yingwen Weng**

Test Description: Test the functionality of the getRank() function in IR_Ballot class. The goal for this is to get the rank of the IR_Ballot object.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created IR_Ballot object.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual function to test the return number of the function to the expected rank	-1	-1	-1	
3					
4					
5					

Post condition(s) for Test:
None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __

Test Date: 2021/3/13

Test Case ID#: IR_Ballot_addRank_1

Name(s) of Testers: Yingwen Weng

Test Description: Test the functionality of the addRank() function in IR_Ballot class. The goal for this is to add the rank of the IR_Ballot object.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created candidate object.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual function to test the return number of the function to the expected rank	addRank(1)	1	1	
3					
4					
5					

Post condition(s) for Test:
None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: IR_Ballot_setRank_1 **Name(s) of Testers: Yingwen Weng**

Test Description: Test the functionality of the setRank() function in IR_Ballot class. The goal for this is to set the rank to the IR_Ballot object on specific position.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created IR_Ballot object.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual function to test the return rank of the function to the expected number	1	1	1	
3					
4					
5					

Post condition(s) for Test:
None

Project Name: Project 1: Voting System **Team# 11**

Test Stage: Unit _1_ System __ **Test Date: 2021/3/13**

Test Case ID#: IR_Ballot_updateRank_1 **Name(s) of Testers: Yingwen Weng**

Test Description: Test the functionality of the updateRank() function in IR_Ballot class. The goal for this is to find the name of the IR_Ballot object.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Automated: yes_1_ no ___

Results: Pass __1__ Fail_____

Preconditions for Test:
Have a created candidate object.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1					
2	Use assertEqual function to test the return rank of the function to the expected number	2	2	2	
3					
4					
5					

Post condition(s) for Test:
None