Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit _X_ System __ **Test Date: 10 November 2023**

Test Case ID#: Intake-Valid-File-Name-001 **Name(s) of Testers: Ruichen He**

Test Description:

Collect user input for a file name.
Read in and store valid file name.

Testing/UnitTest.cpp, uses testing/test_IR.csv

Automated: yes X no

Results: Pass __X__ Fail_____

Preconditions for Test: File name entered is valid

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Compile and run unit test	Testing/UnitTest.cpp Specific Test: IR_Test.intake_valid_file_name_001	Returns: OK	Returns: OK	

Post condition(s) for Test:

A valid file name is accepted. File name is recorded in a variable.

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit _X_ System __ **Test Date: 10 November 2023**

Test Case ID#: Reject-Invalid-File-Name-002 **Name(s) of Testers: Ruichen He**

Test Description:

Collect user input for file name.
Reject invalid file name.

Testing/UnitTest.cpp, uses testing/test_IR.csv

Automated: yes X no ___

Results: Pass X Fail

Preconditions for Test: filename entered is invalid

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
		Testing/UnitTest.cpp Specific Test:	Returns: OK	Returns: OK	
1	Compile and run unit test	IR.Test.reject_invalid_file_name_002			

Post condition(s) for Test:

User is re-prompted for a file name, and no file is recorded.

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit _X_ System __ **Test Date: 10 November 2023**

Test Case ID#: open-file-003 **Name(s) of Testers: Ruichen He**

Test Description:

A file is opened for reading
when given a valid filename.

Testing/UnitTest.cpp, uses testing/test_IR.csv

Automated: yes X no

Results: Pass __X__ Fail_____

Preconditions for Test:

Filename is valid and the file exists.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Compile and run unit test	Testing/UnitTest.cpp Specific Test: IR.Test.open_file_003	Returns: OK	Returns: OK	
2					
3					
4					

Post condition(s) for Test:

File is open for reading and processing.

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit _X_ System __ **Test Date: 10 November 2023**

Test Case ID#: Identify-IR-Vote-Type-004 **Name(s) of Testers: Hannah Nelson**

Test Description:
File indicates vote type is IR
IR election object created

Testing/UnitTest.cpp, uses testing/test_IR.csv

Automated: yes_X_ no ___

Results: Pass __X__ Fail _____

Preconditions for Test: File is opened, formatted correctly, and is type IR ballot

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Compile and run unit test	Testing/UnitTest.cpp Specific Test: IR.Test.identity_IR_vote_type_004	Return OK	Return OK	

Post condition(s) for Test:
IR election is ready to handle next steps

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit _X_ System __ **Test Date: 10 November 2023**

Test Case ID#: Identify-OPL-Vote-Type-005 **Name(s) of Testers: Hannah Nelson**

Test Description:
File indicates vote type is OPL
OPL election object created

Testing/UnitTest.cpp, uses testing/test_OPL.csv

Automated: yes X no ___

Results: Pass X Fail

Preconditions for Test: File is opened, formatted correctly, and is type OPL ballot

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Compile and run unit test	Testing/UnitTest.cpp Specific Test: IR.Test.identity_OPL_vote_type_005	Return OK	Return OK	

Post condition(s) for Test:
OPL election is ready to handle next steps

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit _X_ System __ **Test Date: 10 November 2023**

Test Case ID#: Fair-Coin-Toss-006 **Name(s) of Testers: Ruichen He**

Test Description:

When there is a tie between 2 final candidates, then a fair coin is tossed to decide a winner.

Testing/UnitTest.cpp, uses testing/test_tie_IR.csv

Automated: yes_X__ no __

Results: Pass __X___ Fail_____

Preconditions for Test:

Final candidates have an equal number of votes.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Compile and run test	Testing/UnitTest.cpp Specific Test: IR.Test.fair-coin-toss-006	Returns: OK	Returns: OK	Screenshot of all automatic test in a documentation folder within project 1.

Post condition(s) for Test:

A winner is decided fairly with no bias for position or party.

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit _X_ System __ **Test Date: 10 November 2023**

Test Case ID#: IR-Remove-Lowest-Candidate-007 **Name(s) of Testers: Hannah Nelson**

Test Description:

If there is no candidate with a majority vote and many Candidates are left, then the candidate with the lowest vote percentage is removed.

Test file stored in the "testing" folder. Make file is in the project1 folder.

Automated: yes___ no _X_

Results: Pass _ X ___ Fail _____

Preconditions for Test:

IR voting round is over, multiple candidates are left with no majority.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Run make				
2	Start Election App (./ElectionApp)				
3	Input IR election	testing/test_IR.csv			
4	Lowest candidate is found		Lowest candidate printed to the audit file	Lowest is printed to audit and screen	
5	Votes are cleared from candidate		Votes distributed to other candidates	Candidate removed and votes are redistributed. Can see the candidate is removed in the next round and the votes appear under other candidates	

Post condition(s) for Test:

The candidate with the lowest percentage vote is removed from the candidate pool. Their votes are returned to be redistributed.

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit __ System _X_

Test Date: 10 November 2023

Test Case ID#: IR-Majority-Candidate-008

Name(s) of Testers: Hannah Nelson

Test Description:

A candidate has a majority of the votes and is declared the winner.

Test file stored in the "testing" folder. Make file is in the project1 folder.

Automated: yes no X

Results: Pass X Fail

Preconditions for Test:

End of IR round, votes are distributed.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Run "make"				
2	Run election app (./ElectionApp)				
3	Input IR election	testing/test_IR.csv	Audit file prints winner from who has majority votes in the round	Winner is who has majority of votes	
4					

Post condition(s) for Test:

A winner who has the majority of votes is declared.

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit __ System _X_

Test Date: 12 November 2023

Test Case ID#: IR-Time-Test-009

Name(s) of Testers: Hannah Nelson

Test Description:

A election with over 100,000 ballots runs in under 8 seconds.

Test file stored in the "testing" folder. Make file is in the project1 folder.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:

Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Run "make"				
2	Run election app (./ElectionApp)				
3	Input IR election	testing/timeTest.csv	Winner is declared in under 8 seconds	Winner is declared in under 8 seconds.	
4					

Post condition(s) for Test:

A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit _X_ System __

Test Date: 12 November 2023

Test Case ID#: Audit-File-010

Name(s) of Testers: Hannah Nelson

Test Description:

A election with over 100,000 ballots runs in under 8 seconds.

Test file stored in the "testing" folder. Make file is in the project1 folder.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:

Valid election file.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Run "make"				
2	Run election app (./ElectionApp)				
3	Input IR election	testing/test3_IR.csv	Audit file is produced labeled “test3_IR.csvAudit.txt” and shows ballot placements	Audit file is produced and shows all ballots and their placements	
4					

Post condition(s) for Test:

An audit file is produced which shows which ballots are given to each candidate, how many votes a candidate has, and which candidates are left in the race.

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit __ System _X_

Test Date: 10 November 2023

Test Case ID#: OPL-Winner-011

Name(s) of Testers: Hannah Nelson

Test Description:

Winners are declared using largest remainder formula after filling seats based on quota.

Test file stored in the "testing" folder. Make file is in the project1 folder.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:

End of OPL.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Run "make"				
2	Run election app (./ElectionApp)				
3	Input IR election	testing/test_OPL.csv	Audit file prints winners	Winners are displayed	
4					

Post condition(s) for Test:

A winners are declared for the seats.

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit __ System _X_

Test Date: 12 November 2023

Test Case ID#: OPL-Time-Test-012

Name(s) of Testers: Hannah Nelson

Test Description:

A election with over 100,000 ballots runs in under 8 seconds.

Test file stored in the "testing" folder. Make file is in the project1 folder.

Automated: yes no X

Results: Pass Fail

Preconditions for Test:

Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Run "make"				
2	Run election app (./ElectionApp)				
3	Input IR election	testing/timeOPL.csv	Winner is declared in under 8 seconds		
4					

Post condition(s) for Test:

A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009
Test Description:
A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:
Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declared seconds
4			

Post condition(s) for Test:
A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009
Test Description:
A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:
Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declare seconds
4			

Post condition(s) for Test:
A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009
Test Description:
A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:
Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declared seconds
4			

Post condition(s) for Test:
A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009
Test Description:
A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:
Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declare seconds
4			

Post condition(s) for Test:
A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009
Test Description:
A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:
Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declared seconds
4			

Post condition(s) for Test:
A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009
Test Description:
A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:
Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declare seconds
4			

Post condition(s) for Test:
A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009
Test Description:
A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:
Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declared seconds
4			

Post condition(s) for Test:
A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009
Test Description:
A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:
Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declare seconds
4			

Post condition(s) for Test:
A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009

Test Description:

A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes no X

Results: Pass X Fail

Preconditions for Test:

Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declared seconds
4			

Post condition(s) for Test:

A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009

Test Description:

A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes no X

Results: Pass X Fail

Preconditions for Test:

Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declare seconds
4			

Post condition(s) for Test:

A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009
Test Description:
A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes no X

Results: Pass X Fail

Preconditions for Test:
Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declared seconds
4			

Post condition(s) for Test:
A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System

Test Stage: Unit __ System _X_

Test Case ID#: IR-Time-Test-009
Test Description:
A election with over 100,000 ballots runs in under 8 seconds.

Automated: yes no X

Results: Pass X Fail

Preconditions for Test:
Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result
1	Run "make"		
2	Run election app (./ElectionApp)		
3	Input IR election	testing/timeTest.csv	Winner is declare seconds
4			

Post condition(s) for Test:
A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit __ System _X_ **Test Date: 12 November 2023**

Test Case ID#: IR-Time-Test-009 **Name(s) of Testers: Hannah Nelson**

Test Description:
A election with over 100,000 ballots runs in under 8 seconds.

Test file stored in the "testing" folder. Make file is in the project1 folder.

Automated: yes no X

Results: Pass X Fail

Preconditions for Test:
Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Run "make"				
2	Run election app (./ElectionApp)				
3	Input IR election	testing/timeTest.csv	Winner is declared in under 8 seconds	Winner is declared in under 8 seconds.	
4					

Post condition(s) for Test:
A winner who has the majority of votes is declared in under 8 seconds

Project Name: Project 1: Voting System **Team#12**

Test Stage: Unit __ System _X_

Test Date: 12 November 2023

Test Case ID#: IR-Time-Test-009

Name(s) of Testers: Hannah Nelson

Test Description:

A election with over 100,000 ballots runs in under 8 seconds.

Test file stored in the "testing" folder. Make file is in the project1 folder.

Automated: yes__ no _X__

Results: Pass __X___ Fail_____

Preconditions for Test:

Valid election file contains greater than 100,000 ballots.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Run "make"				
2	Run election app (./ElectionApp)				
3	Input IR election	testing/timeTest.csv	Winner is declared in under 8 seconds	Winner is declared in under 8 seconds.	
4					

Post condition(s) for Test:

A winner who has the majority of votes is declared in under 8 seconds

Project Name: The project #, name of your system, and the team#

Test Stage: Indicate whether it is a unit test or a system test.

Test Date: The date the test was performed.

Test Case ID#: A unique ID is required. Decide on a naming convention and use numbering. Example: Ballot_Shuffle_1

Name(s) of Testers: List the names of anyone involved in running this test case.

Test Description: Describe briefly the test objective.

Automated: Indicate if the test is completely automated or being checked manually. (If you have methods running the tests and checking results, select "yes". If you are manually checking results, indicate manual by selecting the "no.")

Results: Indicate if the test passed or failed.

Step #: You will be listing the test steps in order. This number is the step number in the process.

Test Step Description: Details of the test step.

Test Data: What the test data will be for this step. Be clear on what the input data will be. If using a specific file, be clear on the name.

Expected Result: What result are you expecting from the program component or system.

Actual Result: What result were returned based on the test.

Post condition for Test: What will be true after the test has been run? Has the state of the system changed in any way?

Notes: Comments and notes for you and your team members.