Test logs:

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: C01
Test Description: We create different instances of Candidate class and see if their constructors work well
Name(s) of Testers: Yuanzong Zhang & Zhongyi Sun
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp (labeled Test Candidate Class)
Automated: no
Results: Pass

Preconditions for Test:

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	construct a candidate vector	vector<Candidate> candidates;			No result

2	Instantiate three candidates with different name, and put them into candidate vector	Candidate c1 ("x");
		Candidate c2 ("y");
		Candidate c3 ("z");
					No result

3	Check the candidate name	x, y, z	Candidates names: x y z	Candidates names: x y z	

Post condition(s) for Test: Each candidate can add Ballot and return their number of votes

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: C02
Test Description: We add a ballot to a candidate’s ballots vector and see if it got added successfully
Name(s) of Testers: Yuanzong Zhang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp (labeled Test addBallot)
Automated: no
Results: Pass

Preconditions for Test: Ballot class instance has already been created and has preference map in it.

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Construct an instance of Candidate class	Candidate c_test("Test addBallot");			No result
2	Call addBallot() function	c_test.addBallot(b2);			No result
3	Check the ballot added	1,2,3,4,5,a,b,c,d,e	Ballot 2: 1 a 2 b 3 c 4 d 5 e	Ballot 2: 1 a 2 b 3 c 4 d 5 e	

Post condition(s) for Test: ballot has been successfully assigned to a specific candidate, it can later be used to count a candidate’s total votes or generate report.

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: B01
Test Description: We create different instances of Ballot class and see if their constructors work well
Name(s) of Testers: Yuanzong Zhang & Zhongyi Sun
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp (labeled Test Class Ballot)
Automated: no
Results: Pass

Preconditions for Test:

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	construct a candidate ballot	vector<Ballot> ballots;			No result

2	Instantiate a Ballot	Ballot b1(map1, 1);
		Ballot b2(map2, 1);
					No result

3	Pushback to vector ballot:	ballots.push_back(b1);
		ballots.push_back(b2);
					No result

4	Add candidate name just as the first test case , and their preference to the Ballot	1,2,3,x,y,z 1,2,3,4,5,a,b,c,d,e	Ballot 1: 1 x 2 y 3 z Ballot 2: 1 a 2 b 3 c 4 d 5 e	Ballot 1: 1 x 2 y 3 z Ballot 2: 1 a 2 b 3 c 4 d 5 e	

Post condition(s) for Test: This Ballot can view the candidates’ name and their preference in a map

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: B02
Test Description: We create different instances of Ballot class and see if their get_id() function work well
Name(s) of Testers: Yuanzong Zhang & Zhongyi Sun
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp
Automated: no
Results: Pass

Preconditions for Test: Already construct a ballot vector

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	construct a candidate ballot	vector<Ballot> ballots;			No result

2	Instantiate a Ballot	Ballot b1(map1, 1);
		Ballot b2(map2, 2);
					No result

3	Pushback to vector ballot:	ballots.push_back(b1);
		ballots.push_back(b2);
					No result

4	Add candidate info just as the first test case , and their preference to the Ballot				
5	Check id	1, 2	Ballot ID(inside): 1 Ballot ID(inside): 2	Ballot ID(inside): 1 Ballot ID(inside): 2	

Post condition(s) for Test: Ballot Id can be used when we generate report

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: P01
Test Description: By passing different parameters and to see if Assign_ballot works well
Name(s) of Testers: Zhongyi Sun
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp, assign_ballot()
Automated: no
Results: Pass

Preconditions for Test: Already input the required information which this method use to implement its functionality, and initiate a plurality object

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Initiate a plurality object	Plurality p (ballots, result, input, candidates);			Those input things are from the code in the
2	Call assign_ballot ()	p.assign_ballot();			
3	Check the result print on the termial	temp = p.get_candidates_array(); Print information for temp[0] and temp[1]	x, 3 y, 1	x, 3 y, 1	

Post condition(s) for Test: assign ballot correctly and each candidate has its own ballot list after assigning

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: P02
Test Description: By passing different parameters into the plurality constructor and to see if generate_result works well
Name(s) of Testers: Zhongyi Sun
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp generate_result()
Automated: no
Results: Pass

Preconditions for Test: Already input the required information which this method use to implement its functionality

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Initiate a plurality object	Plurality p (ballots, result, input, candidates);			Those input things are from the code in the
2	Call generate_result()	p.generate_result();			
3	Get winner list	vector<Candidate> winners = result.get_winner_list();			
4	Get loser list	vector<Candidate> losers = result.get_loser_list();			
5	Check the winner and looser		6 winners the x:3 a:2 b:2 y:1	the result is that 6 win the voting they are : x:3 a:2	

			z:1 C:1 2 loosers d:0 e:0	b:2 y:1 z:1 C:1 the result is that 2 loose the voting they are : d:0 e:0	

Post condition(s) for Test: assign ballot correctly and each candidate has its own ballot list after assigning, and we write the detailed information into a “report.txt” file, and print the winner names on the screen.

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: S01
Test Description: By passing different parameters and to see if Assign_ballots works well
Name(s) of Testers: Yuanzong Zhang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp
Automated: no
Results: Pass

Preconditions for Test: Already input the required information which this method use to implement its functionality, and initiate a STV object

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Initiate a plurality object	Plurality p (ballots, result, input, candidates);			Those input things are from the code in the
2	Call assign_ballot ()	p.assign_ballot();			
3	Check the result print on the termial	temp = p.get_candidates_array(); Print information for temp[0] and temp[1]	x, 3 y, 1	x, 3 y, 1	

Post condition(s) for Test: assign ballot correctly and each candidate has its own ballot list after assigning

Project Name: Project 1: Voting System
Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: S02
Test Description: By passing different parameters into the constructor and to see if the generate_report function works well
Name(s) of Testers: Yuanzong Zhang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp
Automated: no
Results: Pass

Preconditions for Test: Already input the required information which this method use to implement its functionality

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Initiate a plurality object	Plurality p (ballots, result, input, candidates);			Those input things are from the code in the
2	Call generate_result()	p.generate_result();			
3	Get winner list	vector<Candidate> winners = result.get_winner_list();			
4	Get loser list	vector<Candidate> losers = result.get_loser_list();			
5	Check the winner and looser		6 winners the x:3 a:2 b:2 y:1 z:1	the result is that 6 win the voting they are : x:3 a:2 b:2	

			C:1 2 loosers d:0 e:0	y:1 z:1 C:1 the result is that 2 loose the voting they are : d:0 e:0	

Post condition(s) for Test: assign ballot correctly and each candidate has its own ballot list after assigning, and we write the detailed information into a “report.txt” file, and print the winner names on the screen.

Test Stage: Unit
Team# 10
Test Date: March 21
Test Case ID#: I01
Test Description: We created instance of Input class and see if its constructor works well
Name(s) of Testers: Yuanzong Zhang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp (labeled Test Input Class)
Automated: no
Results: Pass

Preconditions for Test: The main function got runned and user has input needed information

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Construct an instance of Input class	Input input;			No result

2	Using setter functions in Input class to set values	input.set_num_seats(6);
		input.set_num_ballots(10);
		input.set_num_candidates(8);
		input.set_algorithm(1);
					No result

3	Check the value stored in Input class	10, 8, 6, 1	Number of ballots: 10 Number of candidates: 8 Number of seats: 6 Choose of algorithm: 1	Number of ballots: 10 Number of candidates: 8 Number of seats: 6 Choose of algorithm: 1	

Post condition(s) for Test: store user input information in Input class and can be passed to Plurality or STV class to help processing voting data.
Project Name: Project 1: Voting System
Test Stage: Unit

Team# 10
Test Date: March 21
Test Case ID#: R01
Test Description: We created instance of Result class and see if it successfully stored the result generated by Plurality or STV
Name(s) of Testers: Yuanzong Zhang
Indicate where are you storing the tests (what file) and the name of the method/functions being used: tests/Unit_Test.cpp (labeled Test Result Class)
Automated: no
Results: Pass

Preconditions for Test: The program has finished processing all voting data

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes

1	Create two vectors of Candidate to store winners and losers	vector<Candidate> winners = result.get_winner_list();
		
		vector<Candidate> winners = result.get_loser_list();
					No result

2	Get all ballots assigned to winners	vector<Ballot> ballot_report = winners[i].get_ballots();			No result
3	Check winner list	x: 1,3,4 a: 2,5 b: 7,9 y: 10 z: 6 c: 8	x: 1,3,4 a: 2,5 b: 7,9 y: 10 z: 6 c: 8	x: 1,3,4 a: 2,5 b: 7,9 y: 10 z: 6 c: 8	
4	Check loser list	d, e	d, e	d, e	

Post condition(s) for Test: None

Project Name: Project 1: Voting System
Test Stage: System
Team# 10

Test Date: March 21
Test Case ID#: File01
Test Description: In this test, 8 voters will vote 4 seats out of 11 candidates with Plurality method.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used:
Automated: no
Results: Pass

Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile1.csv			
3	User input the number of seat 4				
4	User input “1” to choose plurality method		Winner should be any four candidates of A, B, C, D, G, K, J and L. Looser should be the rest four of winner list and E, F, H and I.	the result is that 4 win the voting they are : B:1 A:1 G:1 K:1 the result is that 8 loose the voting they are : C:1 D:1 J:1 L:1 E:0 F:0 H:0 I:0	The winner is four random candidates out of A, B, C, D ,G, K, J and L.
5	User input “2”				

	to choose STV method				

Post condition(s) for Test: Winner & Loser information are printed to the screen and a report containing ballots' assignment information is exported to a .txt file.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: March 21

Test Case ID#: File02
Test Description: In this test, 11 voters will vote one seats out of 5 candidates with Plurality method.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used:
Automated: no
Results: Pass

Preconditions for Test:None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile 2.csv			
3	User input the number of seat 4				
4	User input “1” to choose plurality method		Winner should be any onecandidats of C and D. Looser should be the rest one of winner list and B, A and E	the result is that 1 win the voting they are : C:3 the result is that 4 loose the voting they are : D:3 B:2 A:1 E:1	The winner is four random candidates out of C and D.
5	User input “2” to choose STV method				

Post condition(s) for Test:Winner & Loser information are printed to the screen and a report containing ballots’ assignment information is exported to a .txt file.
Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: March 21

Test Case ID#: File03
Test Description: In this test, 22 voters will vote 2 seats out of 15 candidates with Plurality method.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used:
Automated: no
Results: Pass

Preconditions for Test:None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile3.csv			
3	User input the number of seat 4				
4	User input “1” to choose plurality method		Winner should be D and F. Looser should be the rest 13 candidates.	the result is that 2 win the voting they are : D:3 F:3 the result is that 13 loose the voting they are : A:2 H:2 K:2 L:2 O:2 B:1 C:1 G:1 I:1 J:1 N:1 E:0 M:0	

5	User input “2” to choose STV method				

Post condition(s) for Test:Winner & Loser information are printed to the screen and a report containing ballots’ assignment information is exported to a .txt file.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: March 21
Test Case ID#: File04
Test Description: In this test, there are two candidates with same name:”[B]”.

Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used:
Automated: no
Results: Fail

Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile4\(duplicative\candidate\name\).csv	Failed	System does not report error.	This bug is shown in “Buglist”.

Post condition(s) for Test: The program continues to ask user for other input.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: March 21
Test Case ID#: File05
Test Description: In this test, second voters gives two primary ballot to two candidates.
Name(s) of Testers: Ziyi Wang

Indicate where are you storing the tests (what file) and the name of the method/functions being used:
Automated: no
Results: Fail

Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile5\ (duplicative\ ballot\).csv	Failed	System does not report error.	This bug is list in “Buglist”.

Post condition(s) for Test: The program continues to ask user for other input.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: March 21
Test Case ID#: File06
Test Description: In this test, test file is blank file.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used:

Automated: no
Results: Pass

Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile6\(blank\).csv	Program terminate	terminate called after throwing an instance of 'std::out_of_range' what(): basic_string::substr: __pos (which is 1) > this->size() (which is 0) Aborted	

Post condition(s) for Test: Program reports error and terminates.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: March 21
Test Case ID#: File07
Test Description: In this test, each voter has 10 votes, but there is only 6 candidate name.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used:
Automated: no
Results: Pass

Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile7\ (duplicative\ ballot\).csv	Failed	System reports error.	

Post condition(s) for Test: The program reports an error.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: March 21
Test Case ID#: File08
Test Description: In this test, the number of candidate is 6, the ballot number is 4. Each voter has random votes, but the preference is more than the candidates number to test if it will effect the result or not.
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used:
Automated: no
Results: Fail

Preconditions for Test: None

Step #	Test Step Description	TestData	Expected Result	Actual Result	Notes
1	Make file				
2	Load the test file	examplefile8\(\(invalid\ ballot\).csv	Failed	System does not report error.	This bug is list in “Buglist”.

Post condition(s) for Test: Winner & Loser information are printed to the screen and a report containing ballots’ assignment information is exported to a .txt file.

Project Name: Project 1: Voting System
Test Stage: System
Team# 10
Test Date: March 21
Test Case ID#: File09
Test Description: In this test, we try to test different candidate name length. For example using [AB] , rather than A
Name(s) of Testers: Ziyi Wang
Indicate where are you storing the tests (what file) and the name of the method/functions being used:
Automated: no
Results: Fail

Preconditions for Test: None

Step	Test Step	TestData	Expected	Actual Result	Notes

#	Description		Result		
1	Make file				
2	Load the test file	examplefile9(invalid candidates' name).csv			
3	User input the number of seat 5				
4	User input “1” to choose plurality method		Winners are must include D and E, F, but the other two are random	the result is that 5 win the voting they are : D:2 E:1 F:1 A:0 C:0 the result is that 1 loose the voting they are : B:0	We use [AB] as the candidate name, but it only print “A” out.
5	User input “2” to choose STV method				

Post condition(s) for Test: Winner & Loser information are printed to the screen and a report containing ballots’ assignment information is exported to a .txt file.