Test Logs:

Names: Jashwin Acharya (achar061), Carlos Chasi-Mejia(chasi009), Vincent Hoang(hoang317), Steve Petzold(petzo017)

Team: 08

IR Voting Tests:

Unit Tests:

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	**PBI:** As a product owner, I want our voting system to handle the Instant Runoff (IR) Voting election since the Secretary of State asked for it. **Task Description:** Implement and document new unit tests for IR Voting class. **Testing Number: 1**
Team Member(s) Responsible	Jashwin Acharya
Inputs	A CSV file containing all information associated with an election such as Voting algorithm name; list of candidates and party information; number of ballots; number of candidates; and a list of ballots where each ballot contains a ranking of candidates according to voter preference.
Tests	1. Test for when we have 4 candidates out of which 2 are leading and 2 are tied for the lowest votes. After redistribution is performed, one of the two candidates tied for the lowest number of votes is eliminated while the other goes on to win the election. 2. Test for checking that the tie-breaking is fair when two candidates are tied for the lowest number of votes during redistribution and that each candidate has an equal chance of progressing to the next redistribution round. The CSV file used is the same as the first test case above.

Outputs	1. For the first test case, we check if the winner of the election is one of the two candidates who initially were tied for the lowest votes, but end up winning after vote redistribution is complete. 2. For the second test case, we check if the two lowest vote candidates have a 45-55% chance of winning the tie-breaker and eventually winning the election.
Passed or Failed	Passed
Date	12/02/2023

System Tests:

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	**PBI:** As a product owner, I want our voting system to handle the Instant Runoff (IR) Voting election since the Secretary of State asked for it. **Task Description:** Implement and document new system tests for IR Voting class. **Testing Number: 2**
Team Member(s) Responsible	Jashwin Acharya
Inputs	A CSV file containing all information associated with an election such as Voting algorithm name; list of candidates and party information; number of ballots; number of candidates; and a list of ballots where each ballot contains a ranking of candidates according to voter preference.
Tests	1. This test ensures that the user is able to enter the correct name of the CSV file and initiate ballot calculations using the **Instant Runoff (IR) Voting Protocol** where one of the two candidates with the lowest votes after the first round of voting eventually

	goes onto win the whole election after multiple rounds of redistribution. The test also verifies the content of the audit file and the candidate/vote statistics displayed on the screen once ballot calculation is complete.
Outputs	The outputs are the initial prompt displayed on the UI prompting the user to enter the csv file; the audit file for the election; and candidate and vote statistics displayed on the UI once ballot calculation is complete.
Passed or Failed	Passed
Date	12/02/2023

File Parser Tests:

Unit Tests:

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	**PBI:** As an election official, I want the ballot calculations to be as hands-off as possible for ease of use **Task Description:** Implement and document unit tests for MPO file parsing functionality.. **Testing Number: 3**
Team Member(s) Responsible	Carlos Chasi-Mejia
Inputs	A CSV file containing all information of the election such as: voting algorithm name, number of candidates, all candidate names and associated party, number of ballots, number of seats, and list of all ballots. The given file has the prerequisite that all the information in the file is formatted and set up correctly following the write up's example without any inconsistencies.
Tests	1. This test ensures that the correct file header is returned, matching the one in the given test file.

	2. This test ensures that the number of candidates is returned correctly, matching the number given from the test file. 3. This test ensures that all the string containing the candidates and their associated party is returned correctly, matching the line from the given test file. 4. This test ensures that the number of ballots is returned correctly, matching the number given from the test file. 5. This test ensures that the number of seats is returned correctly, matching the number given from the test file. 6. This test ensures that all the ballots from the election are stored in a list correctly, matching all the ballots from the given test file.
Outputs	Ok or error message indicating that the function's output doesn't correctly match the test file's information.
Passed or Failed	Passed
Date	12/02/2023

OPL Voting Tests:

Unit Tests:

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	**PBI:** As a product owner, I want our voting system to handle the Open Party List (OPL) Voting election since the Secretary of State asked for it. **Task Description:** Implement and document new unit tests for OPL Voting class. **Testing Number: 4**

Team Member(s) Responsible	Vincent Hoang
Inputs	A CSV file containing all information associated with an election such as Voting algorithm name; list of candidates and party information; number of ballots; number of seats; number of candidates; and a list of ballots where each ballot contains a 1 that corresponds to a single vote towards a candidate and its party.
Tests	1. Test if the correct winner for a party and candidate is found where a party will only receive a maximum number of seats equal to its number of candidates with no ties occurring in this situation. 2. Test if the correct winner for a party and candidate is found where there are more than two rounds of seat allocations with a tie occurring in this situation.
Outputs	1. For the first test case, we check if the winner of the election is the "D" party with "6" votes towards their party and having "2" seats allocated to them, and the candidate is "Foster" with "4" votes towards them. 2. For the second test case, we check if the winner of the election is the correct party with the correct corresponding candidate, with "3" votes towards their party and "3" seats allocated to them, and "2" votes towards the candidate.
Passed or Failed	Passed
Date	12/01/2023

System Tests:

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	**PBI:** As a product owner, I want our voting system to handle the Open Party List (OPL) Voting election since the Secretary of State asked for it. **Task Description:** Implement and document new system tests for OPL Voting class. **Testing Number: 5**
Team Member(s) Responsible	Vincent Hoang
Inputs	A CSV file containing all information associated with an election such as Voting algorithm name; list of candidates and party information; number of ballots; number of seats; number of candidates; and a list of ballots where each ballot contains a 1 that corresponds to a single vote towards a candidate and its party.
Tests	1. This test ensures that the user is able to enter the correct name of the CSV file and initiate ballot calculations using the **Open Party List (OPL) Voting Protocol** where a party will only receive a maximum number of seats equal to its number of candidates with no ties occurring in this situation. The test also verifies the content of the audit file and the candidate/vote statistics displayed on the screen once ballot calculation is complete. 2. This test ensures that the user is able to enter the correct name of the CSV file and initiate ballot calculations using the **Open Party List (OPL) Voting Protocol** where there are more than two rounds of seat allocations with a tie occurring in this situation. The test also verifies the content of the audit file and the candidate/vote statistics displayed on the screen once ballot calculation is complete.
Outputs	The outputs are the initial prompt displayed

	on the UI prompting the user to enter the csv file; the audit file for the election; and party, candidate, and vote statistics displayed on the UI once ballot calculation is complete.
Passed or Failed	Passed
Date	12/03/2023

MPO Voting Tests:

Unit Tests:

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	**PBI:** As an election official, I want the voting system to handle the Multiple Popularity Only (MPO) voting election since the Secretary of State asked for it and we want the ability to declare multiple winners. **Task Description:** Implement and document unit tests for MPO Voting class. **Testing Number: 6**
Team Member(s) Responsible	Jashwin Acharya
Inputs	A CSV file containing all information associated with an election such as Voting algorithm name; list of candidates and party information; number of seats; number of ballots; number of candidates; and a list of ballots where each ballot contains a 1 only for the candidate and party a person voted for.
Tests	1. This test checks that only one candidate wins a seat since there is only 1 seat available. The candidate who wins the seat also has the highest number of votes. 2. This test ensures that when only 1 seat is available and all candidates are tied with the same number of votes that each candidate has an equal chance of winning the seat. Since we have 6 candidates, each candidate should

	have a 11-21% of winning the seat. 3. This test checks that multiple candidates win seats normally without any type of tie-breakers. 4. This test checks that all candidates win seats after an initial round of tie-breaking is performed between Deutsch and Jones. The winner of the tie breaker should win one seat and the loser's number of votes are compared against Pike's number of seats which eventually leads to Pike and the loser of the inital tie-breaker each winning one seat also. 5. Test for checking if all candidates receive seats after multiple tie breakers are performed. We also ensure that the last candidate, Jai, receives a seat at the end without fighting with another candidate in a tie breaker.
Outputs	1. We assert that the number of votes each candidate received is correct and that only the first candidate, Pike, receives the only seat available for the election. 2. We assert that each candidate has a near-equal chance of winning the election via one tie-breaker since only one seat is available. 3. We assert that the number of votes each candidate received is correct and that Pike, Foster and Jones win seats without any tie-breakers since none of them are tied with the same number of votes. 4. We assert that the number of votes each candidate received is correct and that Pike, Deutsch and Jones all receive seats after an initial round of tie-breaking. 5. We assert that the number of votes each candidate received is correct and that all candidates who received votes win a seat.

Passed or Failed	Passed
Date	12/07/2023

System Tests:

The PBI, the Task Description (from Sprint Log) with Unique Testing Number	**PBI:** As an election official, I want the voting system to handle the Multiple Popularity Only (MPO) voting election since the Secretary of State asked for it and we want the ability to declare multiple winners. **Task Description:** Implement and document unit tests for MPO Voting class. **Testing Number: 7**
Team Member(s) Responsible	Vincent Hoang
Inputs	A CSV file containing all information associated with an election such as Voting algorithm name; list of candidates and party information; number of seats; number of ballots; number of candidates; and a list of ballots where each ballot contains a 1 only for the candidate and party a person voted for.
Tests	1. This test ensures that the user is able to enter the correct name of the CSV file and initiate ballot calculations using the **Multiple Popularity Only (MPO) Voting Protocol** where multiple candidates win seats normally without any type of tie-breakers. The test also verifies the content of the audit file and the candidate/vote statistics displayed on the screen once ballot calculation is complete. 2. This test ensures that the user is able to enter the correct name of the CSV file and initiate ballot calculations using the **Multiple Popularity Only (MPO) Voting Protocol** where all candidates receive seats after multiple

	tie breakers are performed. The test also verifies the content of the audit file and the candidate/vote statistics displayed on the screen once ballot calculation is complete.
Outputs	The outputs are the initial prompt displayed on the UI prompting the user to enter the csv file; the audit file for the election; and party, candidate, and vote statistics displayed on the UI once ballot calculation is complete.
Passed or Failed	Passed
Date	12/09/2023