The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 1
As an election official,
I want the voting system to check for ties for both IR and CPL elections
so that I can determine the final rankings of each candidate.

Task Description (from Sprint Log): TEST_001
Main Task: Resolve the checking for tie error for IRV election.
Task Description: - The voting system should be unbiased and not favor any candidates.
- The voting system should be able to break ties for IRV election based on specifications required in the event of a tie.
- The voting system should be able to analyze and determine the final rankings in IRV election.

Team Member(s) Responsible: Cedric Tan

Inputs:
1. A string representing the election type which is IR.
2. An integer representing the number of candidates.
3. An array list representing the list of candidates with their name and the initial of their party in bracket.
4. An integer representing the number of ballots.
5. An array list representing the ballots received by each candidate.

Tests:
1. Test for inputs where there is a tie occur between two candidates (two candidates have the same number of ballots).
2. Test for inputs where there is a tie occur between three or more candidates (more than two candidates have the same number of ballots).
3. Test for inputs where there is no tie occur between candidates (every candidates have different and unique number of ballots).

Outputs:
True is returned if the checkForTie() function is called.
False is returned if the checkForTie() function is not called.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 1
As an election official,
I want the voting system to check for ties for both IR and CPL elections
so that I can determine the final rankings of each candidate.

Task Description (from Sprint Log): TEST_002
Main Task: Resolve the checking for tie error for CPL election.
Task Description: - The voting system should be unbiased and not favor any candidates.
- The voting system should be able to break ties for CPL election based on specifications required in the event of a tie.
- The voting system should be able to analyze and determine the final rankings in CPL election.

Team Member(s) Responsible: Cedric Tan

Inputs:
1. A string representing the election type which is CPL.
2. An integer representing the number of parties.
3. An array list representing the list of parties' name.
4. An array list representing the list of candidates from the order of each party that are ranked in order.
5. An integer representing the number of seats.
6. An integer representing the number of ballots.
7. An array list representing the ballots received by each party.

Tests:
1. Test for inputs where there is a tie occur between two parties (two parties have the same number of ballots).
2. Test for inputs where there is a tie occur between three or more parties (more than two parties have the same number of ballots).
3. Test for inputs where there is no tie occur between candidates (every parties have different and unique number of ballots).

Outputs:
True is returned if the checkForTie() function is called.
False is returned if the checkForTie() function is not called.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 2
As an election official,
I want the voting system to count ballots for both CPL and IRV election
so that I can determine the total number of ballots for each candidate.

Task Description (from Sprint Log): TEST_003
Main Task: Resolve ballot counting error on IRV election.
Task Description: - There is an issue with CountBallotTest.java for IRV as it did not return the correct ballot counts as expected.
- There might need a bigger array size for ballot count variable.
- The voting system should be able to count all ballots from each ballot file accurately.
- The voting system should be able to identify the number of ballots for each candidate.

Team Member(s) Responsible: Cedric Tan

Inputs:
1. A string representing the election type which is IR.
2. An integer representing the number of candidates.
3. An array list representing the list of candidates with their name and the initial of their party in bracket.
4. An integer representing the number of ballots.
5. An array list representing the ballots received by each candidate.

Tests:
1. Test for inputs where the election type is IR and the ballots are valid.
2. Test for inputs where the election type is OPL, an invalid election type and the ballots are valid.
3. Test for inputs where the election type is IR and the ballots are invalid.
4. Test for inputs where the election type is OPL, an invalid election type and the ballots are invalid.

Outputs:
OK message is returned if the expected total number of ballots is equal to the actual total number of ballots. Otherwise, an error message indicating that the number of ballots is out of bounds is returned. (The system will ignore invalid ballots and continue to count the number of valid ballots.)

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 2
As an election official,
I want the voting system to count ballots for both CPL and IRV election
so that I can determine the total number of ballots for each candidate.

Task Description (from Sprint Log): TEST_004
Main Task: Resolve ballot counting error on CPL election.
Task Description:
- There is an issue with CountBallotTest.java for CPL as it did not return the correct ballot counts as expected.
- There might need a bigger array size for ballot count variable.
- The voting system should be able to count all ballots from each ballot file accurately.
- The voting system should be able to identify the number of ballots for each party.

Team Member(s) Responsible: Bryan Lee

Inputs:
1. A string representing the election type which is CPL.
2. An integer representing the number of parties.
3. An array list representing the list of parties' name.
4. An array list representing the list of candidates from the order of each party that are ranked in order.
5. An integer representing the number of seats.
6. An integer representing the number of ballots.
7. An array list representing the ballots received by each party.

Tests:
1. Test for inputs where the election type is CPL and the ballots are valid.
2. Test for inputs where the election type is OPL, an invalid election type and the ballots are valid.
3. Test for inputs where the election type is CPL and the ballots are invalid.
4. Test for inputs where the election type is OPL, an invalid election type and the ballots are invalid.

Outputs:
OK message is returned if the expected total number of ballots is equal to the actual total number of ballots. Otherwise, an error message indicating that the number of ballots is out of bounds is returned. (The system will ignore invalid ballots and continue to count the number of valid ballots.)

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 3
As an election official,
I want the voting system to count ballots for CPL election
so that I can determine the total number of ballots for each party.

Task Description (from Sprint Log): TEST_005
Main Task: Unit Testing for countBallot() function
Task Description:
- The feature must be able to count all valid ballots effeciently without any syntax or runtime errors.
- The feature must be able to allocate valid ballots to each party.
- The feature must be integrated with the rest of the voting system and is able to work as expected.
- The feature must be thoroughly tested and must pass all test cases in both Unit Testing and Integration Testing.
- The feature must meet all the acceptance criterias.

Team Member(s) Responsible: Bryan Lee

Inputs:
1. A string representing the election type which is CPL.
2. An integer representing the number of parties.
3. An array list representing the list of parties' name.
4. An array list representing the list of candidates from the order of each party that are ranked in order.
5. An integer representing the number of seats.
6. An integer representing the number of ballots.
7. An array list representing the ballots received by each party.

Tests:
1. Test for inputs where the election type is CPL and the ballots are valid.
2. Test for inputs where the election type is OPL, an invalid election type and the ballots are valid.
3. Test for inputs where the election type is CPL and the ballots are invalid.
4. Test for inputs where the election type is OPL, an invalid election type and the ballots are invalid.

Outputs:
OK message is returned if the ballots are valid. Otherwise, a file containing the invalidated ballots is returned.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 4
As an election official,
I want the voting system to determine the rankings of each party for an CPL election
so that I can determine the winner of the CPL election.

Task Description (from Sprint Log): TEST_006
Main Task: Unit Testing for checkRanking() function for CPL election
Task Description: - The feature can calculate the rankings of the candidates effeciently without any syntax or runtime errors.
- The feature can identify ties in the rankings and calls finalRanking.java in the event of a tie.
- The feature should be integrated with the rest of the voting system and is able to generate the rankings of all candidates.
- The code should be clearly documented and easy to follow.
- The feature should be tested and must pass all test cases in both Unit Testing and Integration Testing.
- The feature must meet all the acceptance criterias.

Team Member(s) Responsible: Bryan Lee

Inputs:
1. A string representing the election type which is CPL.
2. An integer representing the number of parties.
3. An array list representing the list of parties' name.
4. An array list representing the list of candidates from the order of each party that are ranked in order.
5. An integer representing the number of seats.
6. An integer representing the number of ballots.
7. An array list representing the ballots received by each party.

Tests:
1. Test for inputs where the election type is CPL with number of ballots of each party.

Outputs:
OK message is returned if the expected rankings is equal to the actual rankings. Otherwise, an error message indicating that the rankings was failed to be determined is returned.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 5
As an election official,
I want the voting system to determine the rankings of the candidates for an IRV election
so that I can determine the winner of the IRV election.

Task Description (from Sprint Log): TEST_007
Main Task: Unit Testing for checkRanking() function for IRV election
Task Description: - The feature can calculate the rankings of the candidates effeciently without any syntax or runtime errors.
- The feature can identify ties in the rankings and calls finalRanking.java in the event of a tie.
- The feature should be integrated with the rest of the voting system and is able to generate the rankings of all candidates.
- The code should be clearly documented and easy to follow.
- The feature should be tested and must pass all test cases in both Unit Testing and Integration Testing.
- The feature must meet all the acceptance criterias.

Team Member(s) Responsible: Sherryl Ooi

Inputs:
1. A string representing the election type which is IR.
2. An integer representing the number of candidates.
3. An array list representing the list of candidates with their name and the initial of their party in bracket.
4. An integer representing the number of ballots.
5. An array list representing the ballots received by each candidate.

Tests:
1. Test for inputs where the election type is IR with number of ballots of each candidate along with their name.

Outputs:
OK message is returned if the expected rankings is equal to the actual rankings. Otherwise, an error message indicating that the rankings was failed to be determined is returned.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 6
As an election official,
I want to the voting system to generate an audit file with the election information at a given time
so that I can download it for verification and auditing purposes.

Task Description (from Sprint Log): TEST_008
Main Task: Unit Testing for generateAuditFile() function
Task Description: - The feature must generate audit files efficiently without any syntax or runtime errors.
- The code must be clearly documented and easy to follow.
- The contents within the generated audit file must adhere to a given format as the specifications required and must be accurate.
- The feature should be integrated with the rest of the voting system and is working as expected.
- The feature must be thoroughly tested and must pass all test cases in both Unit Testing and Integration Testing.
- The feature must meet all the acceptance criterias.

Team Member(s) Responsible: Sherryl Ooi

Inputs:
Case 1: IR
1. A string representing the election type which is IR.
2. An integer representing the number of candidates.
3. An array list representing the list of candidates with their name and the initial of their party in bracket.
4. An integer representing the number of ballots.
5. An array list representing the ballots received by each candidate.

Case 2: CPL
1. A string representing the election type which is CPL.
2. An integer representing the number of parties.
3. An array list representing the list of parties' name.
4. An array list representing the list of candidates from the order of each party that are ranked in order.
5. An integer representing the number of seats.
6. An integer representing the number of ballots.
7. An array list representing the ballots received by each party.

Tests:
1. Test for inputs where the election type is IR and the election is completed.
2. Test for inputs where the election type is CPL and the election is completed.

Outputs:
OK message is returned if the audit file is successfully being generated. Otherwise, an error message indicating that the audit file failed to be generated is returned.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number: PBI: 7 As an election official, I want the voting system to be capable of performing Popularity Only (PO) voting so that the voting system has more flexibility in conducting elections. Task Description (from Sprint Log): **Testing Number N/A** Main Task: Develop function to run PO election in the voting system Task Description: - The voting system should be able to determine a single winner based on the candidate receiving the most ballots with a fair coin toss if there is a tie or ties between candidates accurately. - The voting system should be able to perform PO voting when reading the ballot file.
Team Member(s) Responsible: N/A
Inputs: N/A
Tests: This function is not developed in the voting system. Hence, there is no test case made.
Outputs: N/A
Passed or Failed: N/A
Date: April 30, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number: PBI: 8 As an election official, I want the voting system to be capable of bringing in PO election ballots via a .csv file so that the voting system has more flexibility in conducting elections. Task Description (from Sprint Log): **Testing Number N/A** Main Task: Develop a function that brings in PO election ballots via a .csv file in the voting system to run PO elections. Task Description: - The voting system should be able to bring in PO election ballots via a .csv file.
Team Member(s) Responsible: N/A
Inputs: N/A
Tests: This function is not developed in the voting system. Hence, there is no test case made.
Outputs: N/A
Passed or Failed: N/A
Date: April 30, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number: PBI: 9 As an election official, I would like the voting system to be able to read mulitple files during an election so that I can bring in different files from different balloting locations. Task Description (from Sprint Log): TEST_009 Main Task: Develop function to read multiple files Task Description: - Modify existing readFile() function in fileSystem.java class to take in multiple flies with the correct file format. - The voting system should be able to read multiple ballot files in the correct file format. - The voting system should ensure that all ballot files being brought into the system is not modified. - The voting system should be able to check if there are any write-in candidates in each ballot file.
Team Member(s) Responsible: Bryan Lee
Inputs: 1. Multiple ballot files are selected by the user during an election.
Tests: 1. Test for input where the user selects multiple ballot files to be read in during an election. 2. Test for input where the user selects only one ballot file to be read in during an election.
Outputs: OK message is returned if the ballot files are successfully being read in. Otherwise, an error message indicating that the system fails to read in the ballot files is returned.
Passed or Failed: Passed
Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 10
As an election official,
I would like the voting system to bring in the election files with the option to input the date of the election so that I can manage the election efficiently without the need for manual data entry.

Task Description (from Sprint Log): TEST_010
Main Task: Develop option to input date of election
Task Description:
- Implement a GUI window in fileSystem.java for users to input date of election after bringing in election files.
- The voting system should be able to import the needed information to run the election.
- The graphical interface of the feature should be user-friendly and easily accessible.
- The voting system should be able to handle unexpected data inputs and alert users for any inconsistencies.

Team Member(s) Responsible: Sherryl Ooi

Inputs:
1. Prompt user option (Yes or No) whether to input the date of the election when bringing in the election files into the voting system.

Tests:
1. Test for input where the user chooses Yes to input the date of the election when bringing in the election files into the voting system.
2. Test for input where the user chooses No to not input the date of the election when bringing in the election files into the voting system.

Outputs:
If Yes is chosen, the system will ask user to input the date of the election when bringing in the election files into the voting system.
If No is chosen, the system will not ask user to input the date of the election when bringing in the election files into the voting system.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 11
As an election official,
I would like the voting system to allow users to select ballot files
so that I can select specific files from a directory for the voting system to process.

Task Description (from Sprint Log): TEST_011
Main Task: Develop function to select ballot files from a directory
Task Description: - Implement a GUI window in fileSystem.java which allows users to select specific files from a directory for file processing.
- The voting system should include a window that appears at the start of the program.
- The window should allow users to type in a file name or look for file(s) on disk using their mouse or arrow keys.
- The graphical interface of the feature should be user-friendly and easily accesible so that users can easily navigate through their directory.

Team Member(s) Responsible: Bryan Lee

Inputs:
1. Prompt user option (Yes or No) whether to select ballot files from a directory.

Tests:
1. Test for input where the user chooses Yes to select ballot files from a directory.
2. Test for input where the user chooses No to not select ballot files from a directory.

Outputs:
If Yes is chosen, the system will ask user to select a ballot file from a directory.
If No is chosen, the system will not ask user to select a ballot file from a directory.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number: PBI: 12 As an election official, I would like the voting system to be able to process election information from mulitple files during an election so that I can consolidate the election data for multiple ballot files. Task Description (from Sprint Log): TEST_012 Main Task: Modify readFile() function to process multiple ballot files Task Description: - Modify the existing readFile() function to be able to integrate election data from multiple ballot files. - The voting system should be able to integrate election data from multiple ballot files. - The voting system should be able to check if the contents within each ballot file adheres to specifications.
Team Member(s) Responsible: Cedric Tan
Inputs: 1. Prompt user option (Yes or No) whether to read in multiple ballot files during an election.
Tests: 1. Test for input where the user chooses Yes to read in multiple ballot files during an election. 2. Test for input where the user chooses No to stop reading in multiple ballot files during an election.
Outputs: If Yes is chosen, the system will ask user to select a ballot file to be read in. If No is chosen, the system will not ask user to select a ballot file to be read in.
Passed or Failed: Passed
Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 13
As an election official,
I want the voting system to validate the IRV ballots based on the requirements of the state election officials so that the voting system can remove any invalidated ballots and store them in a separate file for audit purposes.

Task Description (from Sprint Log): TEST_013
Main Task: Develop function for IRV ballots validation
Task Description:
- Implement a function that validates IRV ballots based on the requirementss of the state election officials and store invalidated ballots in a separate file for audit purposes.
- The voting system should be modified to be able to validate IRV ballots based on updated specifications.
- The voting system should identify and remove any invalidated ballots to preserve the accuracy and integrity of the election results.
- The voting system should then store the invalidated ballots into a separate file for audit purposes.

Team Member(s) Responsible: Sherryl Ooi

Inputs:
1. A string representing the election type which is IR.
2. An integer representing the number of candidates.
3. An array list representing the list of candidates with their name and the initial of their party in bracket.
4. An integer representing the number of ballots.
5. An array list representing the ballots received by each candidate.

Tests:
1. Test for inputs where the election type is IR and the ballots are valid.
2. Test for inputs where the election type is OPL, an invalid election type and the ballots are valid.
3. Test for inputs where the election type is IR and the ballots are invalid.
4. Test for inputs where the election type is OPL, an invalid election type and the ballots are invalid.

Outputs:
OK message is returned if the ballots are valid. Otherwise, a file containing the invalidated ballots is returned.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 14
As an election official,
I want the voting system to be capable of performing both IRV and CPL types of voting
so that the voting system has more flexibility in conducting elections.

Task Description (from Sprint Log): TEST_014
Main Task: Develop IRV and CPL voting
Task Description: - The IRV and CPL algorithms are used in the voting system to run elections separately.
- The voting system should be able to handle and run both elections accurately.
- The voting system should be able to differentiate between IRV and CPL types of voting when reading the ballot file.

Team Member(s) Responsible: Cedric Tan

Inputs:
Case 1: IR
1. A string representing the election type which is IR.
2. An integer representing the number of candidates.
3. An array list representing the list of candidates with their name and the initial of their party in bracket.
4. An integer representing the number of ballots.
5. An array list representing the ballots received by each candidate.

Case 2: CPL
1. A string representing the election type which is CPL.
2. An integer representing the number of parties.
3. An array list representing the list of parties' name.
4. An array list representing the list of candidates from the order of each party that are ranked in order.
5. An integer representing the number of seats.
6. An integer representing the number of ballots.
7. An array list representing the ballots received by each party.

Tests:
1. Test for inputs where the election type is IR and the election is completed.
2. Test for inputs where the election type is CPL and the election is completed.

Outputs:
OK message is returned if the system is able to recognize both IR and CPL successfully via .csv files being brought in. Otherwise, an error message indicating that the system fails to recognize the election type is returned.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 14
As an election official,
I want the voting system to be capable of performing both IRV and CPL types of voting
so that the voting system has more flexibility in conducting elections.

Task Description (from Sprint Log): TEST_015
Main Task: System Testing for IRV and CPL voting system
Task Description: - The feature can run both election types effeciently without any syntax or runtime errors.
- The feature must be integrated with the rest of the voting system and works as the specifications required.
- The code should be clearly documented and easy to follow.
- The feature should be thoroughly tested and must pass all test cases in both Unit Testing and Integration Testing.
- The feature must meet all the acceptance criterias.

Team Member(s) Responsible: Bryan Lee

Inputs:
Case 1: IR
1. A string representing the election type which is IR.
2. An integer representing the number of candidates.
3. An array list representing the list of candidates with their name and the initial of their party in bracket.
4. An integer representing the number of ballots.
5. An array list representing the ballots received by each candidate.

Case 2: CPL
1. A string representing the election type which is CPL.
2. An integer representing the number of parties.
3. An array list representing the list of parties' name.
4. An array list representing the list of candidates from the order of each party that are ranked in order.
5. An integer representing the number of seats.
6. An integer representing the number of ballots.
7. An array list representing the ballots received by each party.

Tests:
1. Test for inputs where the election type is IR and the election is completed.
2. Test for inputs where the election type is CPL and the election is completed.

Outputs:
OK message is returned if the system is able to recognize both IR and CPL successfully via .csv files being brought in. Otherwise, an error message indicating that the system fails to recognize the election type is returned.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 15
As an election official,
I want the voting system to display a table illustrating each round of the IRV election
so that I can see the change in the number of votes for each round.

Task Description (from Sprint Log): TEST_016
Main Task: Develop function to display tables showing each round of IRV election
Task Description:
- Modify displayResults.java to ensure that it works as the updated specifications required.
- The voting system should display a table illustrating the number of votes that the candidate added / subtracted for each round.
- The voting system should display the table in an easily readible format to show a round-by-round analysis of votes.
- The table should accurately reflect the results of the IRV election as implemented by the voting system.

Team Member(s) Responsible: Sherryl Ooi

Inputs:
1. A string representing the election type which is IR.
2. An integer representing the number of candidates.
3. An array list representing the list of candidates with their name and the initial of their party in bracket.
4. An integer representing the number of ballots.
5. An array list representing the ballots received by each candidate.

Tests:
1. Test for inputs where the election type is IR and the election is completed.

Outputs:
OK message is returned with a table illustrating each round of the IRV election is displayed where the change in the number of votes for each round can be seen. Otherwise, an error message indicating that the table is failed to be generated is returned.

Passed or Failed: Passed

Date: April 29, 2023

The PBI, the Task Description (from Sprint Log) with Unique Testing Number:
PBI: 16
As an election official,
I want the voting system to be capable of displaying the PO stats at the end of the election
so that I can know the percentage of votes each candidate received.

Task Description (from Sprint Log): **Testing Number N/A**
Main Task: Develop a function that calculates the PO stats at the end of the election in the voting system.
Task Description: - The voting system should be able to display PO stats to the screen at the end of the election.
- The PO stats must contain information regarding the percentage of votes each candidate received so that constituents who votes know how well their candidate did in the election.
- The PO stats should contain information about who won and who lost.

Team Member(s) Responsible: N/A

Inputs:
N/A

Tests:
This function is not developed in the voting system. Hence, there is no test case made.

Outputs:
N/A

Passed or Failed: N/A

Date: April 30, 2023