Product Backlog

Team 4: Bryan Lee, Cedric Tan, Sherryl Ooi

Product 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.	
Note	- There is an error within checkForTie() function in finalRanking class. - The system is not checking for tie and did not prompt the user to run fair/pool coin toss in the event of a tie as stated in the buglist.
Acceptance Criteria	- The voting system should be unbiased and not favor any candidates or parties. - The voting system should be able to break ties for both IR and CPL elections based on specifications required in the event of a tie. - The voting system should be able to analyze and determine the final rankings in both IR and CPL elections.
Definition of Done	- The feature can identify ties in both IR and CPL elections efficiently without any syntax or runtime errors. - The feature can identify whether to use a pool coin toss or fair coin toss when there's a tie between multiple candidates/parties. - The feature must be integrated with the rest of the voting system and is able to generate the final rankings of all candidates/parties. - The code must be clearly documented and easy to follow. - 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.
Effort	Medium
PBI Author	Cedric Tan

Product 2	
As an election official I want the voting system to count ballots for IRV election so that I can determine the total number of ballots for each candidate.	
Note	- There is an error within countBallot() function in countBallot class. - The ballot counting for IRV is having error as stated in the buglist. - 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.
Acceptance Criteria	- 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.
Definition of Done	- 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 candidate. - 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.
Effort	Small
PBI Author	Cedric Tan

Product 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 candidate.	
Note	- There is an error with the unit testing for countBallot() function — countBallotTest.java. - The ballot counting for CPL does not have any errors. - 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.
Acceptance Criteria	- 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.
Definition of Done	- 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.
Effort	Small
PBI Author	Bryan Lee

Product 4	
As an election official I want the voting system to determine the rankings of each candidate and party for an CPL election so that I can determine the winner of the CPL election.	
Note	- There is an error with the unit testing for checkRanking() function — rankingsTest.java. - The system fails the rankingTest for CPL during testing because it did not return the correct rankings of each candidate as stated in the buglist. - There is an issue with the rankings array.
Acceptance Criteria	- The voting system should be able to calculate the rankings of all candidates and parties based on the number of votes received. - The voting system should be able to determine the winner of the election based on the rankings
Definition of Done	- 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.
Effort	Small
PBI Author	Bryan Lee

Product 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.	
Note	- There is an error with the unit testing for checkRanking() function — rankingsTest.java. - The system fails the rankingTest for IRV during testing because it did not return the correct rankings of each candidate as stated in the buglist. - There is an issue with the rankings array.
Acceptance Criteria	- The voting system should be able to calculate the rankings of all candidates and parties based on the number of votes received. - The voting system should be able to determine the winner of the election based on the rankings.
Definition of Done	- 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.
Effort	Small
PBI Author	Sherryl Ooi

Product 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.	
Note	- There is an error with the unit testing for generateAuditFile() function — displayResultsTest.java. - There is an issue with displayResultsTest.java during our testing as it did not return the correct GUI as expected based on the buglist. - There is an error displaying the JFrame and JOptionPane.
Acceptance Criteria	- The voting system should be able to generate an audit file which serves as a clear record of all votes cast to verify the accuracy of the election results. - The contents within the audit file should be clearly documented to ensure the transparency and integrity of the election results. - The generated audit file should be in a readible file format and can be downloaded.
Definition of Done	- 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.
Effort	Small
PBI Author	Sherryl Ooi

Product 7	
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.	
Note	- The modification of the existing readFile() function in fileSystem.java class.
Acceptance Criteria	- 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.
Definition of Done	- The feature should be modified to read multiple ballot files effeciently without any syntax or runtime errors. - The feature can identify whether the ballot files are in .csv format and displays an error message when the file format is incorrect. - The feature should keep looping and prompt the user to upload another file. - The feature must be integrated with the rest of the voting system and work as expected. - The code must be clearly documented and easy to follow. - 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.
Effort	Small
PBI Author	Bryan Lee

Product 8	
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.	
Note	- The implementation of an additional feature in fileSystem.java which includes a GUI window.
Acceptance Criteria	- 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.
Definition of Done	- The feature should allow users to input date of election effeciently without any syntax or runtime errors. - The feature can identify whether the inputted date of electino is correct and displays an error message when there are any inconsistencies. - The GUI for inputting date of the election must adopt a user-centric design approach and must be integrated into the user interface of the voting system. - The code must be clearly documented and easy to follow. - 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.
Effort	Small
PBI Author	Sherryl Ooi

Product 9	
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.	
Note	- The implementation of an additional feature in fileSystem.java which includes a GUI window.
Acceptance Criteria	- 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.
Definition of Done	- The feature should allow users to select ballot files easily without any syntax or runtime errors. - The GUI for selecting ballot files must adopt a user-centric design approach and must be integrated into the user interface of the voting system. - The code must be clearly documented and easy to follow. - 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.
Effort	Small
PBI Author	Bryan Lee

Product 10	
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.	
Note	- The implementation of an additional feature in fileSystem.java.

Acceptance Criteria	- 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.
Definition of Done	- The feature should be modified to process multiple ballot files effeciently without any syntax or runtime errors. - The feature should loop through the first line inside ballot files to determine election type. - The feature should then loop through the other lines inside ballot files to store and update the election information. - The voting system should be able to differentiate between valid and invalid ballots. - The feature must be integrated with the rest of the voting system and process each file correctly. - The code must be clearly documented and easy to follow. - 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.
Effort	Small
PBI Author	Cedric Tan

Product 11	
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.	
Note	- The implementation of an additional feature in readFile.java.
Acceptance Criteria	- 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.
Definition of Done	- The feature is modified to identify invalidated ballots effeciently without any syntax or runtime errors. - The feature must have at least half of the candidates ranked for the ballot to be valid. - The candidates' rank must be rounded up from .5 or above to the next higher integer value. - 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 must create a file which stores the invalidated ballots titled "invalidated_dateofelection.xxx". - The feature should be thoroughly tested and must pass all test cases in both Unit Testing and Integration Testing.
Effort	Small
PBI Author	Sherryl Ooi

Product 12	
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.	
Note	- The IRV and CPL algorithms are used in the voting system to run elections separately.
Acceptance Criteria	- 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.
Definition of Done	- 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.
Effort	Small
PBI Author	Cedric Tan

Product 13	
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.	
Note	- Modify displayResults.java to ensure that it works as the updated specifications required.
Acceptance Criteria	-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.
Definition of Done	- The feature must be modified to display the table correctly without any syntax or runtime errors. - The code must be clearly documented and easy to follow. - The table must adhere to a given table format as the specifications required and the information 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.
Effort	Medium

PBI Author	Sherryl Ooi