PBI #1

As an election official,
I need to be able to run a popularity-only election
because the Secretary of State has determined that it is a valid election type for the state.

Acceptance Criteria:

1. Given a CSV file in the popularity-only format, audit and report files are produced.
2. The audit file contains the date and time, election type, number of candidates, candidates, number of ballots, ballots, percentages of votes per each candidate, and winner.
3. The report file contains the date and time, election type, number of candidates, candidates, number of ballots, percentages of votes per each candidate, and winner.
4. If there are multiple candidates with the highest number of votes, then a candidate must be randomly chosen from those candidates.

Definition of Done:

1. All acceptance criteria are met.
2. All acceptance tests that can be automated are automated.
3. All acceptance tests are met.
4. Test logs have been created for all related tests.
5. Documentation has been created for all related components.
6. The feature is able to be demoed.
7. The changes have been accepted.
8. Duplicate code has been removed through refactoring
9. Messy and poorly-designed code has been cleaned up through refactoring as per our Java style guide

Effort: Large

PBI Author(s): Nikunj Chawla

PBI #2

As an election official,
I need to be able to bring in the ballots for a popularity-only election
so that a winner can be determined for the popularity-only election type.

Acceptance Criteria:

1. CompuVote inputs files that are formatted as specified for the popularity-only election.
2. CompuVote parses files that are formatted as specified for the popularity-only election.
    a. CompuVote correctly imports the election type as the popularity-only election type
    b. CompuVote correctly imports the number of candidates
    c. CompuVote correctly imports the candidates
    d. CompuVote correctly imports the number of ballots
    e. CompuVote correctly imports each of the ballots

3. CompuVote prints an error and exits if the provided popularity-only election file has an issue in formatting
Definition of Done: 1. All acceptance criteria are met. 2. All acceptance tests that can be automated are automated. 3. All acceptance tests are met. 4. Test logs have been created for all related tests. 5. Documentation has been created for all related components. 6. The feature is able to be demoed. 7. The changes have been accepted. 8. Duplicate code has been removed through refactoring 9. Messy and poorly-designed code has been cleaned up through refactoring as per our Java style guide
Effort: Small
PBI Author(s): Nikunj Chawla

PBI #3
As an election official, I want to be able to see the statistics about the popularity-only election so that constituents can know how well candidates did in the election.
Acceptance Criteria: 1. The candidates, number of votes received, and the percentages of votes they each received should be displayed on the screen for the popularity-only election, preferably in a table format.
Definition of Done: 1. All acceptance criteria are met. 2. All acceptance tests that can be automated are automated. 3. All acceptance tests are met. 4. Test logs have been created for all related tests. 5. Documentation has been created for all related components. 6. The feature is able to be demoed. 7. The changes have been accepted. 8. Duplicate code has been removed through refactoring 9. Messy and poorly-designed code has been cleaned up through refactoring as per our Java style guide
Effort: Small
PBI Author(s): Nikunj Chawla

PBI #4

As an election official,
I would like to bring in ballots from multiple balloting locations to run the election
so that all ballots for an election are processed together from different polling locations.

Acceptance Criteria:

1. CompuVote is able to accept multiple files as input.
2. CompuVote is able to parse all the files as one election, only parsing one file's pre-ballot information (i.e. election type, number of candidates, candidates, and number of seats if applicable)

Definition of Done:

1. All acceptance criteria are met.
2. All acceptance tests that can be automated are automated.
3. All acceptance tests are met.
4. Test logs have been created for all related tests.
5. Documentation has been created for all related components.
6. The feature is able to be demoed.
7. The changes have been accepted.
8. Duplicate code has been removed through refactoring
9. Messy and poorly-designed code has been cleaned up through refactoring as per our Java style guide

Effort: Extra Large

PBI Author(s): Aaron Kandikatla, Nikunj Chawla

PBI #5

As an election official,
I want to be able to provide files by some visual or graphics-related method
so it is easier to use compared to using the terminal or command line

Acceptance Criteria:

1. CompuVote allows users to look at file(s) on the disk using their mouse or arrow keys through a GUI.
2. The GUI allows users to select and unselect files.
3. The GUI displays the files in a list in the order chosen by the user.
4. The GUI allows the user to remove and add files until they are satisfied with their choice in file selection.

Definition of Done:

1. All acceptance criteria are met.
2. All acceptance tests that can be automated are automated.
3. All acceptance tests are met.
4. Test logs have been created for all related tests.
5. Documentation has been created for all related components.
6. The feature is able to be demoed.
7. The changes have been accepted.

8. Duplicate code has been removed through refactoring
9. Messy and poorly-designed code has been cleaned up through refactoring as per our Java style guide

Effort: Extra Large

PBI Author(s): Aaron Kandikatla

PBI #6

As an election official,
I want the IR election system to invalidate ballots that do not rank at least half of the candidates
so that the requirement by state election officials is met.

Acceptance Criteria:
1. CompuVote removes all ballots that do not rank at least half of the candidates in an instant-runoff voting system from consideration.

Definition of Done:
1. All acceptance criteria are met.
2. All acceptance tests that can be automated are automated.
3. All acceptance tests are met.
4. Test logs have been created for all related tests.
5. Documentation has been created for all related components.
6. The feature is able to be demoed.
7. The changes have been accepted.
8. Duplicate code has been removed through refactoring
9. Messy and poorly-designed code has been cleaned up through refactoring as per our Java style guide

Effort: Medium

PBI Author(s): Aaron Kandikatla

PBI #7

As an election official,
I want to be able to see the ballots that were invalidated for the IR election system
so that they can be reviewed for auditing purposes.

Acceptance Criteria:
1. Each ballot that has been invalidated in the IR election system will be printed to a file with the name of “invalidated_[dateofelection].txt” without quotes, replacing [dateofelection] with the date/time in the format “[yyyy]-[MM]-[dd]_[HH]-[mm]-[ss]”, replacing [yyyy] with the four-digit year, [MM] with the 2-digit month (adding 0s as

necessary), [dd] with the 2-digit day of the month (adding 0s as necessary), [HH] with the 2-digit hour in 24-hour time (adding 0s as necessary), [mm] with the 2-digit minutes (adding 0s as necessary), and [ss] with the 2-digit seconds (adding 0s as necessary)
2. The ballot information printed will include the ballot number and ranked candidates in order of ranking.

Definition of Done:
1. All acceptance criteria are met.
2. All acceptance tests that can be automated are automated.
3. All acceptance tests are met.
4. Test logs have been created for all related tests.
5. Documentation has been created for all related components.
6. The feature is able to be demoed.
7. The changes have been accepted.
8. Duplicate code has been removed through refactoring
9. Messy and poorly-designed code has been cleaned up through refactoring as per our Java style guide

Effort: Medium

PBI Author(s): Jack Fornaro

PBI #8

As an election official,
I would like to see the distribution of votes and the change in vote numbers after each round for the IR election system
so that I can see how the election generally progressed.

Acceptance Criteria:
1. CompuVote prints the following information in a table:
    a. The candidates
    b. The candidates' respective parties
    c. The candidates' votes for each round
    d. The exhausted pile of votes for each round
    e. The total votes only under the first round
    f. The change in votes for each candidate in each round
    g. The change in votes for the total only under the first round
    h. The change in votes for the exhausted pile for each round
    i. distribution of votes and change in vote numbers after each round into a table.
2. The table is shown on the summary on the screen.
3. The summary correctly displays the distribution of votes and change in vote numbers.

Definition of Done:
1. All acceptance criteria are met.
2. All acceptance tests that can be automated are automated.

3. All acceptance tests are met.
4. Test logs have been created for all related tests.
5. Documentation has been created for all related components.
6. The feature is able to be demoed.
7. The changes have been accepted.
8. Duplicate code has been removed through refactoring
9. Messy and poorly-designed code has been cleaned up through refactoring as per our Java style guide

Effort: Medium

PBI Author(s): Jack Fornaro

PBI #9

As an election official,
I would like to provide the winners and the losers to the election officials
so they can validate the election.

Acceptance Criteria:

1. CompuVote outputs a report file.
2. The date of the election is printed to the report file.
3. The type of election is printed to the report file.
4. The candidates and their parties are printed to the report file.
5. The number of seats is printed to the report file if applicable.
6. The winner of the election is printed to the report file.

Definition of Done:

1. All acceptance criteria are met.
2. All acceptance tests that can be automated are automated.
3. All acceptance tests are met.
4. Test logs have been created for all related tests.
5. Documentation has been created for all related components.
6. The feature is able to be demoed.
7. The changes have been accepted.
8. Duplicate code has been removed through refactoring
9. Messy and poorly-designed code has been cleaned up through refactoring as per our Java style guide

Effort: Small

PBI Author(s): Jack Fornaro