

<p>As a developer</p> <p>I want Open Party Election to distinguish a winner between a tie of more than two candidates</p> <p>So that a final winner can be chosen fairly</p>
<p>Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete)</p> <ul style="list-style-type: none"> <li>• A singular winner is determined correctly when three or more candidates are in a tie</li> </ul>
<p>Definition of Done (what is required by the team before sending out for review)</p> <ul style="list-style-type: none"> <li>• No errors are thrown</li> <li>• The correct winner is displayed</li> <li>• Bug has been resolved and buglisted updated</li> <li>• Test log is updated</li> </ul>
<p>Effort: Medium</p>
<p>PBI Author(s)</p> <ul style="list-style-type: none"> <li>• Andrew</li> </ul>

<p>As an election official</p> <p>I want OPL Elections to properly allocate seats in any situation</p> <p>So that elections are fair and accurate</p>
<p>Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete)</p> <ul style="list-style-type: none"> <li>• Seat allocation works when more parties than seats</li> <li>• Results displayed accurately reflect election outcome</li> </ul>
<p>Definition of Done (what is required by the team before sending out for review)</p> <ul style="list-style-type: none"> <li>• OPL Election is properly documents</li> <li>• Unit tests are updated</li> <li>• Test and bug logs are updated</li> <li>• OPL election code runs without any bugs</li> </ul>
<p>Effort: Medium</p>
<p>PBI Author(s)</p> <ul style="list-style-type: none"> <li>• Nikhil</li> </ul>

<p>As tester/developer</p> <p>I want the OPL party unit tests to pass all cases</p> <p>So that we know the class methods are fully functional</p>
---

<p>Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete)</p> <ul style="list-style-type: none"> <li>• All tests pass</li> <li>• Bug has been eliminated</li> </ul>
<p>Definition of Done (what is required by the team before sending out for review)</p> <ul style="list-style-type: none"> <li>• File is properly documented</li> <li>• Test cases are added to the testing log</li> <li>• Bug log is updated.</li> </ul>
<p>Effort: Medium</p>
<p>PBI Author(s)</p> <ul style="list-style-type: none"> <li>• Nikhil</li> </ul>

<p>As an Election Official</p> <p>I want an election that determines a single winner based on a candidate receiving the most balls</p> <p>So that popularity only election results can be determined</p>
<p>Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete)</p> <ul style="list-style-type: none"> <li>• PO Elections can be run and produce results</li> <li>• Ties are resolved with a fair coin flip</li> </ul>
<p>Definition of Done (what is required by the team before sending out for review)</p> <ul style="list-style-type: none"> <li>• Tests pass</li> <li>• Documentation complete</li> <li>• </li> </ul>
<p>Effort:</p> <ul style="list-style-type: none"> <li>• Large</li> </ul>
<p>PBI Author(s)</p> <ul style="list-style-type: none"> <li>• Alex</li> </ul>

<p>As an Election Official</p> <p>I want to bring in PO ballots through a .csv file</p> <p>So that PO elections can be run</p>
<p>Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete)</p> <ul style="list-style-type: none"> <li>• PO election is run with a .csv input</li> <li>• PO election successfully runs</li> </ul>
<p>Definition of Done (what is required by the</p>

team before sending out for review)

- Unit and Integration tests are complete
- Documentation is added accordingly
- Testing log and bug log are updated
- Code compiles

Effort: Small

PBI Author(s);

- Alex
- Nikhil

As an Election Official

I want to see statistics displayed at the end of a PO election

So that I know who won and lost.

Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete)

- Each candidate shows percentage of votes at election termination

Definition of Done (what is required by the team before sending out for review)

- Unit and Integration tests are complete
- Documentation is added accordingly
- Testing log and bug log are updated
- Code compiles

Effort: Small

PBI Author(s);

- Alex
- Nikhil

As an election official

I want to be able to run elections with multiple ballot files

So that election results from more than one location can be accounted for

Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete)

- Program can be run with 2 or more ballot files
- Results with 2 or more ballot files are accurate and fair

Definition of Done (what is required by the team before sending out for review)

- Unit and Integration tests are complete
- Documentation is added accordingly

- Testing log and Bug log are updated
- Code compiles

Effort: Large

PBI Author(s):

- Nikhil

As the user, I need a GUI to prompt me for the filename, so that the process is more intuitive and any election official could use it

Acceptance Criteria:

- A GUI has been created
- The user can use the GUI to grab the correct ballot file

Definition of Done:

- GUI code is complete
- GUI code is fully documented to doxygen standards
- Unit and integration tests are developed, and code passes to satisfaction

Effort: Large

PBI Author(s): Peter Linden

As a local election official, I need a method of certifying that ballots are valid, and be able to remove invalid ballots from counting so that election results are valid

Acceptance Criteria:

- Any ballots in which less than half of candidates (round up from 0.5) are ranked are labeled as invalid
- Any ballots labeled as invalid are excluded from election counting

Definition of Done:

- Code compiles
- Documentation makes sense
- Unit tests are created and passed to our satisfaction

Effort: Medium

PBI Author(s): Peter Linden

As a local election official, I need a method of storing the ballot data for all invalidated ballots for audit purposes

Acceptance Criteria:

- A file with the format <invalidated\_dateofelection.xxx> has been created

<ul style="list-style-type: none"> <li>- All invalidated ballots from the current election will be added to the above file</li> </ul>
Definition of Done: <ul style="list-style-type: none"> <li>- File can be created correctly</li> <li>- Documentation of code is enumerated</li> <li>- Unit testing of edge cases is developed and passed</li> </ul>
Effort: Small
PBI Author(s): Peter Linden

As an election official I would like the handling of IR elections to include a table of results at the end so that the overall movement of votes can be tracked visually as per the IR methodology
Acceptance Criteria: <ul style="list-style-type: none"> <li>- A table is displayed after the election is counted</li> <li>- This table displays the changes for each round, i.e. the votes removed and added from/to each candidate</li> </ul>
Definition of Done: <ul style="list-style-type: none"> <li>- Code functionality is complete</li> <li>- Documentation of code is up to standards</li> <li>- Unit testing is fully implemented, and code passes parameters</li> </ul>
Effort: Medium (based off existing data, just needs effort to be formatted correctly)
PBI Author(s): Peter Linden

As an election official I would like an additional post-election report that I can give to election certification officials so that they can certify the basics of the election
Acceptance Criteria: <ul style="list-style-type: none"> <li>- Post-election report is generated</li> <li>- Report contains all the listed requirements: date, type of election, candidates # of seats, and winners.</li> </ul>
Definition of Done: <ul style="list-style-type: none"> <li>- Code is properly documented</li> <li>- Unit tests to test different election scenarios are created and passed to our satisfaction</li> </ul>
Effort: Small
PBI Author(s): Peter Linden