

PBI	Task	Hours	Who	Status
<p>As an election official, I want ties to be handled fairly so the decision is unbiased when running an OPL election. AC: Ties in OPL are decided randomly. Done: Tested, documentation done, source code updated to reflect this change, reviewed and refactored. Effort: Small (30 minutes)</p>	Code OPL ties	1	Bek	Done
	Design OPL tie test cases	1	Bek	Done
	Execute OPL tie test cases	0.5	Bek	Done
	Refactor OPL (if necessary)			Done
<p>As an election official, I want the program to run smoothly because I don't know how to handle program crashes. AC: On any input the program exits gracefully, notifying the user of any error in layman's terms. Done: Tested, documents, and source code updated to reflect this change, reviewed and refactored. Effort: Small</p>	Fix divide by zero bug in OPL	1	Logan	Done
	Execute no ballot test case	0.25	Logan	Done
<p>As an election official, I want to choose the winner(s) based upon who has received the most votes because the Secretary of State has decided this is a fair way to choose winners. AC: The winner(s) is/are chosen based upon who received the most votes in one round. The program can be run on CSE lab machines. Done: Implemented, tested, documented, reviewed and refactored. Effort: M</p>	Design MPO voting system (UML etc)	1	Perrie	Done
	Implement MPOVotingSystem class	2	Perrie	Done
	Design unit test cases for MPOVotingSystem class	1	Matthew	Done
	Execute unit test cases for MPOVotingSystem class	0.5	Matthew	Done
	Document MPOVotingSystem class	0.5	Matthew	Done
<p>As an election official, I need ties between candidates to be broken by a coin toss so that the election is fair and accurate. AC: The coin toss leads to each candidate winning the given seat an even amount of time. Done: Implemented, tested, documented, reviewed and refactored Effort: S</p>	Implement MPO tie breaking functionality	1	Logan	Done
	Design unit test cases for tie breaking functionality	1	Logan	Done
	Execute unit test cases for tie breaking functionality	0.5	Logan	Done
	Document tie breaking functionality	0.5	Logan	Done
<p>As an election official, I need the program to be able to handle files of the MPO type so that the system can process the results automatically. AC: File is read in and processes all election information and assigns ballots to candidates. Done: Implemented, tested, documented, reviewed and refactored. Effort: S</p>	Update Election class to handle MPO	1	Perrie	Done
	Update HeaderProcessor class to handle MPO	1	Matthew	Done
	Update FileProcessor to work with MPO voting system	1	Matthew	Done
	Manually test MPO file parsing (test files)	1	Bek	In Progress
	Update Election class documentation	0.5	Perrie	Done
	Update HeaderProcessor class documentation	0.5	Matthew	Done
	Update FileProcessor class documentation	0.5	Matthew	Done
	Design unit tests for Election class updates	1	Perrie	Done
	Execute unit tests for Election class updates	0.5		Not Started
	Design unit tests for HeaderProcessor class updates	1	Bek	In Progress
	Execute unit tests for HeaderProcessor class updates	0.5	Bek	Not Started
	Design unit tests for FileProcessor class updates	1	Bek	In Progress
	Execute unit tests for FileProcessor class updates	0.5	Bek	Not Started
	System integration testing for MPO voting system file handling	2	Logan	Done
	Update MPOVotingSystem to output stats showing percentages when the election is complete.	2	Matthew	Done
<p>As an election official, I need the stats from the MPO election to be outputted to the screen once the election is decided. AC: Each candidate has their statistics displayed to the screen upon completion of the election. Can run on CSELabs machines. Done: Implemented, tested, and documented. Effort: Small</p>	Unit test MPOVotingSystem stats output.	1	Perrie	Not Started
	Update MPOVotingSystem documentation to reflect added ability to output stats.	1	Matthew	Done
	Update FileProcessor to disqualify IRV Ballots that do not have >50% of candidates ranked.	1	Logan	Done
<p>As an election official, I want ballots with less than half of the candidates ranked in an IRV election to be discounted and noted in a file. AC: Program invalidates ballots with less than half of the candidates ranked and reports which ballots were invalidated to the audit file. Done: Implemented, tested, documented, reviewed and refactored Effort: Medium</p>	Update FileProcessor documentation	0.5	Logan	Done
	Unit test FileProcessor on an IRV election with ballots that do not have >50% of candidates ranked.	1	Perrie	Not Started