

Committed Backlog Items	NOT STARTED	IN PROGRESS	COMPLETED
<p>CPL multiple files</p> <p>User Story: As an election official, I would like the system to support the importing of multiple files for the CPL (Closed Party List) voting system, enabling us to bring in different files from various balloting locations to facilitate a streamlined voting process.</p> <p>Acceptance Criteria: The system must allow the selection and simultaneous importing of multiple files specifically formatted for the CPL voting system. It must ensure the compatibility and proper integration of data from these files into the CPL voting process, processing all ballots without fail. The system must validate the integrity and format of the imported files to ensure accurate CPL voting results.</p> <p>Effort: Large</p>		Task4: System test	Task1: Unit tests Task2: Code Task3: Update Documentation
<p>OPL multiple files</p> <p>User Story: As an election official, I need the system to facilitate the importing of multiple files for the OPL (Open Party List) voting system, allowing for the inclusion of various files from different balloting locations to enhance the OPL voting process's comprehensiveness and accuracy.</p> <p>Acceptance Criteria: The system must support the selection and simultaneous importing of multiple files, tailored for the OPL voting system. It should guarantee the seamless integration of data from these files into the OPL voting process, ensuring no ballot is overlooked. The system is required to check the integrity and format of the imported files for accurate OPL results.</p> <p>Effort: Large</p>		Task4: System test	Task1: Unit tests Task2: Code Task3: Update Documentation

<p>MPO files(in general)</p> <p>User Story: As an election official, I need the system to have the capability to import MPO (Multiple Popularity Only) election ballots from a .csv file.</p> <p>Acceptance Criteria: The system must allow the importation of ballot data from a .csv file format. The import function should be able to handle .csv files containing the names of candidates and the corresponding votes they have received. The system must validate the format and integrity of the .csv file to prevent errors during the election process.</p> <p>Effort: Large</p>		Task4: System test	Task1: Unit tests Task2: Code Task3: Update Documentation
<p>MPO multiple files</p> <p>User Story: As an election official, I need the system to facilitate the importing of multiple files for the MPO (Multiple Popularity Only) voting system, allowing the inclusion of various files from different balloting locations to efficiently process and tally votes from a broader geographic area.</p> <p>Acceptance Criteria: The system must support the selection and simultaneous importing of multiple files designed for the MPO voting system. It should ensure seamless integration of data from these files into the MPO voting process, guaranteeing that every ballot is processed. The system must validate the integrity and format of the imported files to ensure the accuracy of the MPO election results.</p> <p>Effort: Large</p>		Task4: System test	Task1: Unit tests Task2: Code Task3: Update Documentation

<p>MPO Election</p> <p>User Story: As the election official, I would like to run a MPO. So that I can get multiple winners from those candidates that receive the most ballots.</p> <p>Acceptance Criteria: The system can determine the total number of winners needed based on the specified number of seats. In the event of a tie, the system must decide the winner(s) through a fair coin toss process.(fair and unbiased)</p> <p>Effort: Medium</p>		<p>Task4: System test</p>	<p>Task1: Unit tests Task2: Code Task3: Update Documentation</p>
<p>MV files(in general)</p> <p>User Story: As an election official, I need the system to support the importation of Municipal Voting (MV) election ballots from a .csv file.</p> <p>Acceptance Criteria: The system must support the importation of MV ballot data from a .csv file format. The import function should successfully process .csv files containing the voters' choices, ensuring the data matches the expected format of candidate names and vote counts. The system must validate the integrity and format of the .csv file to ensure the data is accurately reflected in the voting results without errors. Upon successful import, the system should verify and summarize the ballot data, including a count of votes for each candidate and any discrepancies detected.</p> <p>Effort: Medium</p>		<p>Task4: System test</p>	<p>Task1: Unit tests Task2: Code Task3: Update Documentation</p>

<p>MV multiple files</p> <p>User Story: As an election official, I require the system to support the importing of multiple files for the MV (Municipal Voting) system, enabling the efficient management and processing of ballots from various polling locations.</p> <p>Acceptance Criteria: The system must allow the selection and importing of multiple files simultaneously, catering specifically to the MV voting format. The system should ensure seamless integration of ballot data from these files into the MV voting process, accurately processing every ballot. It must validate the integrity and format of the imported files to prevent any discrepancies in the MV election results. The system should provide election officials with a detailed summary of the import process, highlighting any issues with specific files to promote transparency and ease of troubleshooting.</p> <p>Effort: Large</p>		Task4: System test	Task1: Unit tests Task2: Code Task3: Update Documentation
<p>MV election</p> <p>User Story: As an election official, I need an algorithm, termed Municipal Voting (MV), that allows determining multiple winners based on the number of ballots received by each candidate.</p> <p>Acceptance Criteria: The system must accurately count votes per candidate and identify if a tie occurs. In the event of a tie, the system performs a fair coin toss to determine the winner among the tied candidates. The system allows voters to select from 1 up to the total number of candidates.</p> <p>Effort: Large</p>		Task4: System test	Task1: Unit tests Task2: Code Task3: Update Documentation

<p>MPO stats to screen</p> <p>User Story: As an election official, I want the MPO (Multiple Popularity Only) voting process to display detailed election stats at the end of the election, including the percentage of votes each candidate received. .</p> <p>Acceptance Criteria: At the end of the MPO voting process, the system displays the election results on the screen. The results include the percentage of votes each candidate received. The results clearly indicate which candidates won seats and which did not. The display of results is designed to be easily understandable by election officials and the public.</p> <p>Effort: Medium</p>		<p>Task4: System test</p>	<p>Task1: Unit tests Task2: Code Task3: Update Documentation</p>
<p>MV stats to screen</p> <p>User Story: As an election official, I require the system to display Municipal Voting (MV) stats at the end of the election, showing clear information on who won and who lost, and detailing the number of votes received by all candidates.</p> <p>Acceptance Criteria: The information should be presented in an easily understandable format, suitable for immediate interpretation by election officials and the public. The system should ensure the accuracy and integrity of the displayed stats, reflecting the true outcome of the election.</p> <p>Effort: Medium</p>		<p>Task3: Update Documentation Task4: System test</p>	<p>Task1: Unit tests Task2: Code</p>