

SELECTED BACKLOG ITEMS	DEV NAME	TASKS NOT STARTED	TASKS IN PROGRESS	TASKS COMPLETED
<p>As an election official  <b>I want</b> to be able to process an STV-style election correctly  <b>So that</b> when the Secretary of State announces an election of that type, winners can be determined.</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"> <li>- STV can handle any number of seats and candidates where the number of seats is less than or equal to the number of candidates, but greater than zero.</li> <li>- Ballot shuffling is toggleable and functions correctly</li> <li>- STV accurately produces the winners and losers of the election</li> </ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"> <li>- Code refactoring</li> <li>- Exceptions are handled gracefully</li> <li>- In-line documentation is updated</li> <li>- Unit and integration tests are written and pass</li> </ul> <p><b>Effort:</b></p> <ul style="list-style-type: none"> <li>- Small</li> <li>- 2-3 Hours</li> </ul> <p><b>PBI Author(s):</b></p> <ul style="list-style-type: none"> <li>- Baanee Singh, Josh Subhan</li> </ul>	<p>Amelia Lunning</p> <p>Amelia Lunning</p> <p>Amelia Lunning</p> <p>Baanee Singh</p>	<p>Adjust unit and system tests for Main accordingly</p> <p>Adjust documentation as necessary</p>	<p>Add additional unit and system tests for STV</p>	<p>Fix bug where running STV with 1-2 seats results in an OutOfMemory error</p>
<p>As an election official  <b>I want</b> to be able to run elections without having to manually input any parameters into the system except for the election csv file  <b>So that</b> erroneous or fraudulent information is not given, and elections are secure.</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"> <li>- System should successfully parse the following from the file: <ul style="list-style-type: none"> <li>- Election type</li> <li>- Number of seats</li> <li>- Number of candidates</li> <li>- Number of ballots</li> <li>- Candidate names</li> </ul> </li> <li>- CSV file cannot be modified</li> <li>- CSV file cannot undergo any preprocessing</li> </ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"> <li>- Documentation (JavaDocs, In-line comments)</li> <li>- Code refactoring</li> <li>- Unit and integration tests are written and pass</li> <li>- All exceptions are handled gracefully</li> </ul> <p><b>Effort:</b></p> <ul style="list-style-type: none"> <li>- Medium</li> <li>- 3-4 hours</li> </ul> <p><b>PBI Author(s):</b> Josh Subhan</p>	<p>Baanee Singh</p> <p>Baanee Singh</p> <p>Baanee Singh</p> <p>Baanee Singh</p> <p>Baanee Singh</p> <p>Josh Subhan</p>		<p>Edit methods that will parse all info from the headers.</p> <p>Create new unit and system tests (Scanner object as a parameter)  Implement new main() logic</p> <p>Refactoring Main</p>	<p>Documentation in-line</p> <p>Documentation javadocs</p>
<p>As an election official  <b>I want</b> to be able to process ballots that are designed for a municipal voting style election  <b>So that</b> when the Secretary of State announces an election of that type, ballots can be appropriately counted and winners determined</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"> <li>- Election winners are properly determined</li> <li>- Accurately count ballots where a voter has only voted for one candidate</li> <li>- Accurately count ballots where a voter has voted for more than one candidate</li> <li>- Accurately count ballots where a voter has voted for all candidates</li> <li>- Settle a tie between candidates with a fair coin toss mechanism</li> </ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"> <li>- Documentation (JavaDocs, In-line comments).</li> <li>- Unit and integration tests are written and pass</li> <li>- All exceptions are handled gracefully</li> </ul> <p><b>Effort:</b></p> <ul style="list-style-type: none"> <li>- Large</li> <li>- 5-6 hours</li> </ul> <p><b>PBI Author(s):</b> Josh Subhan</p>	<p>Zach Larsen</p> <p>Zach Larsen</p> <p>Zach Larsen</p> <p>Zach Larsen</p> <p>Zach Larsen</p> <p>Zach Larsen</p> <p>Zach Larsen</p> <p>Zach Larsen</p> <p>Zach Larsen</p> <p>Baanee Singh</p> <p>Baanee Singh</p> <p>Baanee Singh</p> <p>Josh Subhan</p> <p>Josh Subhan</p> <p>Josh Subhan</p> <p>Josh Subhan</p>	<p>Create system tests for MV class</p> <p>Write out logs for tests</p>	<p>Unit test tabulateVotes()</p> <p>Unit test determine_Winner_Loser()</p> <p>Unit test settleTies()</p> <p>Create JavaDocs for MV class</p> <p>Create JavaDocs for MV testing class</p> <p>Create in-line comments for MV class</p> <p>Refactor abstract Election class</p>	<p>Create MV class</p> <p>Create MV testing class</p> <p>Code MV constructor, setDeterministic, startElection methods</p> <p>Code tabulateVotes() method</p> <p>Code determine_Winner_Loser() method</p> <p>Code settleTies() method</p> <p>Refactor plurality and MV inheritance structure</p> <p>Create new DirectSelection abstract class</p>
<p>As an election official  <b>I want</b> to be able to bring in multiple files from different balloting locations for both STV and Plurality voting  <b>So that</b> the election data can be processed and managed efficiently from different locations.</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"> <li>- System must be able to accept multiple csv files when STV or PV are selected as the election type</li> <li>- System must be able to stop accepting files when prompted by the user</li> <li>- System properly collects all the information from ballots in each csv file</li> </ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"> <li>- Refactoring.</li> <li>- Unit and integration tests are written and pass</li> <li>- All exceptions are handled gracefully</li> <li>- Documentation (JavaDocs, In-line comments)</li> </ul> <p><b>Effort:</b></p> <ul style="list-style-type: none"> <li>- Medium</li> <li>- 4-5 Hours</li> </ul> <p><b>PBI Author(s):</b>  Baanee Singh</p>	<p>Josh Subhan</p> <p>Baanee Singh</p> <p>Baanee Singh</p> <p>Josh Subhan</p> <p>Josh Subhan</p>	<p>Create unit and integration tests</p> <p>Documentation in-line</p> <p>Documentation javadocs</p>		<p>Create a function to handle multiple file inputs</p> <p>Create logic to continue to prompt the user for more files or to stop prompting</p>