

# **Project 2: Election Tabulation System**

**Team #10**

**Testing Logs**

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Fully functional and correct STV implementation <b>Task:</b> Fix OutOfMemory Bug for 1-2 Seats <b>Testing Number:</b> 1
<b>Team Member(s) Responsible:</b>	Amelia Lunning
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- STV election configuration with 1-2 seats</li> <li>- Sample ballots for testing</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- Run STV with 1 seat and verify no OutOfMemory error occurs.</li> <li>- Run STV with 2 seats and verify no OutOfMemory error occurs.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- STV runs without memory errors for 1-2 seats.</li> <li>- Correct winners are determined.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Fully functional and correct STV implementation <b>Task:</b> Add STV Unit/System Tests <b>Testing Number:</b> 2
<b>Team Member(s) Responsible:</b>	Amelia Lunning Baanee Singh
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- Existing STV test suite</li> <li>- New test cases for edge scenarios (e.g., tied votes, exhausted ballots)</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- Verify STV handles tied votes correctly.</li> <li>- Verify STV handles exhausted ballots gracefully.</li> <li>- Verify STV with ballot shuffling enabled.</li> <li>- Verify STV eliminates candidates in the correct order.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- New unit and system tests added to the test suite.</li> <li>- All tests pass.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Fully functional and correct STV implementation <b>Task:</b> Adjust Main Tests <b>Testing Number:</b> 3
<b>Team Member(s) Responsible:</b>	Amelia Lunning
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- Existing Main test suite</li> <li>- Updated STV logic</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- Verify Main correctly initializes STV election with 1-2 seats.</li> <li>- Verify Main handles STV results display accurately.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- Updated Main tests pass.</li> <li>- No regressions in existing functionality.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Fully functional and correct STV implementation <b>Task:</b> Adjust documentation as necessary <b>Testing Number:</b> 4
<b>Team Member(s) Responsible:</b>	Baane Singh
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- Current documentation</li> <li>- Updated STV logic and test cases</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- Verify in-line comments reflect changes.</li> <li>- Verify user-facing documentation is updated.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- Documentation is accurate and up-to-date.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Fully functional and correct STV implementation <b>Task:</b> Refactor genAudit() and displayResults() <b>Testing Number:</b> 5
<b>Team Member(s) Responsible:</b>	Amelia Lunning
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- Current genAudit() and displayResults() methods</li> <li>- Refactoring requirements (e.g., readability, efficiency)</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- Verify genAudit() produces correct audit logs.</li> <li>- Verify displayResults() outputs results accurately.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- Refactored methods are cleaner and more efficient.</li> <li>- Functionality remains unchanged.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Process file header with election information <b>Task:</b> Edit methods that will parse all info from the headers <b>Testing Number:</b> 6
<b>Team Member(s) Responsible:</b>	Baanee Singh Josh Subhan
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- CSV file with headers (election type, seats, candidates, ballots, candidate names).</li> <li>- Scanner object reading file.</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- Verify correct election type extraction.</li> <li>- Validate number of seats parsed.</li> <li>- Check candidate names are read accurately.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- Parsed election parameters stored.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Process file header with election information <b>Task:</b> Documentation Javadocs & In-Line <b>Testing Number:</b> 7
<b>Team Member(s) Responsible:</b>	Baanee Singh
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- Current documentation</li> <li>- Updated Main logic and test cases</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- Verify in-line comments reflect changes.</li> <li>- Verify user-facing documentation is updated.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- Documentation is accurate and up-to-date.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Process file header with election information <b>Task:</b> Refactor Main() & Implement new main() logic <b>Testing Number:</b> 8
<b>Team Member(s) Responsible:</b>	Baanee Singh Josh Subhan
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- CSV file path</li> <li>- CSV file with headers (election type, seats, candidates, ballots, candidate names).</li> <li>- Scanner object reading file.</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- System test with valid file.</li> <li>- Graceful handling of malformed files.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- Election results printed to console/log.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task</b>	<b>PBI:</b> Process file header with election information
--------------------------	---

<b>Description (from Sprint Log) with Unique Testing Number:</b>	<b>Task:</b> Write Unit/Integration Tests <b>Testing Number:</b> 9
<b>Team Member(s) Responsible:</b>	Baanee Singh Josh Subhan
<b>Inputs:</b>	- Mock CSV files (valid/invalid headers).
<b>Tests:</b>	- End-to-end parsing test. - Verify error messages for invalid inputs.
<b>Outputs:</b>	- New unit and system tests added to the test suite. - All tests pass.
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> MV algorithm <b>Task:</b> Create MV Class <b>Testing Number:</b> 10
<b>Team Member(s) Responsible:</b>	Zach Larsen
<b>Inputs:</b>	- Create MV class extending DirectSelection. - Declare fields: candidates, ballots, isDeterministic.
<b>Tests:</b>	- Verify MV extends DirectSelection.
<b>Outputs:</b>	- MV.java file with class skeleton.
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> MV algorithm <b>Task:</b> Implement MV Constructor <b>Testing Number:</b> 11
--	--

<b>Team Member(s) Responsible:</b>	Zach Larsen
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- Election CSV file path.</li> <li>- Initialize MV with CSV parsing logic (reuse DirectSelection superclass).</li> <li>- Add validation for municipal election type.</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- Check object creation.</li> <li>- Verify exception handling.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- Functional constructor.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> MV algorithm <b>Task:</b> Code setDeterministic() Method <b>Testing Number:</b> 12
<b>Team Member(s) Responsible:</b>	Zach Larsen
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- Boolean flag (true for testing, false for production).</li> <li>- Implement method to toggle deterministic coin toss for ties.</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- No need to test since it just switches between true and false and is implemented into other methods.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- Method to control tie-breaking randomness.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> MV algorithm <b>Task:</b> Implement startElection() <b>Testing Number:</b> 13
--	---

<b>Team Member(s) Responsible:</b>	Zach Larsen
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- Loaded ballots from CSV.</li> <li>- Validate ballot format (municipal-specific rules).</li> <li>- Call tabulateVotes() and determineWinnerLoser().</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- Test the complete election process, verifying that winners are selected and an audit file is created.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- Election results and audit file generated and deleted.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> MV algorithm <b>Task:</b> determineWinnerLoser() <b>Testing Number:</b> 14
<b>Team Member(s) Responsible:</b>	Zach Larsen
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- Vote counts from tabulateVotes().</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- Rank candidates by votes.</li> <li>- Identify ties for settleTies().</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- Ordered list of candidates (winners first).</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> MV algorithm <b>Task:</b> Implement settleTies() <b>Testing Number:</b> 15
<b>Team Member(s) Responsible:</b>	Zach Larsen



<b>Inputs:</b>	- List of tied candidates.
<b>Tests:</b>	- Use Random with seed if isDeterministic=true. - Log coin toss results.
<b>Outputs:</b>	- Resolved winner(s) from tied candidates.
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> MV algorithm <b>Task:</b> Refactor Inheritance <b>Testing Number:</b> 16
<b>Team Member(s) Responsible:</b>	Zach Larsen Josh Subhan
<b>Inputs:</b>	- None. (Abstract Class)
<b>Tests:</b>	- Create DirectSelection abstract class (parent of MV and Plurality). - Move common logic (e.g., CSV parsing) to DirectSelection.
<b>Outputs:</b>	- Updated class hierarchy.
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> MV algorithm <b>Task:</b> System Testing <b>Testing Number:</b> 17
<b>Team Member(s) Responsible:</b>	Zach Larsen
<b>Inputs:</b>	- MV Input Structure

<b>Tests:</b>	- System test for MV that covers all methods
<b>Outputs:</b>	- Successful run of the MV election
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> MV algorithm <b>Task:</b> Documentation <b>Testing Number:</b> 18
<b>Team Member(s) Responsible:</b>	Baanee Singh Josh Subhan
<b>Inputs:</b>	- Current documentation - Updated logic and test cases
<b>Tests:</b>	- Write JavaDocs for all MV methods. - Add in-line comments for complex logic. - Update README.md with MV election rules.
<b>Outputs:</b>	- JavaDocs, README updates.
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Bring in Multiple Files for Both STV and Plurality <b>Task:</b> Create a function to handle multiple file inputs <b>Testing Number:</b> 19
<b>Team Member(s) Responsible:</b>	Josh Subhan Baanee Singh
<b>Inputs:</b>	- List of CSV file paths (user-provided).
<b>Tests:</b>	- Tests with a single valid CSV filename. - Test multiple valid CSV filenames.

	<ul style="list-style-type: none"> <li>- Test single invalid file extension.</li> <li>- Test multiple invalid filenames.</li> <li>- Test valid file has already been entered.</li> </ul>
<b>Outputs:</b>	- Errors for invalid; Pass for Valid
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Bring in Multiple Files for Both STV and Plurality <b>Task:</b> Implement User Prompt Logic <b>Testing Number:</b> 20
<b>Team Member(s) Responsible:</b>	Josh Subhan
<b>Inputs:</b>	- User input (yes/no via Scanner).
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- Test for continue on yes.</li> <li>- Test for termination on no.</li> </ul>
<b>Outputs:</b>	- File loop either continues or terminates on command.
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Bring in Multiple Files for Both STV and Plurality <b>Task:</b> Refactor Main tests & Create unit and integration tests <b>Testing Number:</b> 21
<b>Team Member(s) Responsible:</b>	Josh Subhan
<b>Inputs:</b>	- Based on any edge cases that come up.
<b>Tests:</b>	- Ensure tests account for new functionalities and code/tests didn't break once those new methods.

<b>Outputs:</b>	- All tests pass with little to no refactoring.
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025

<b>The PBI, the Task Description (from Sprint Log) with Unique Testing Number:</b>	<b>PBI:</b> Bring in Multiple Files for Both STV and Plurality <b>Task:</b> Documentation & README Update <b>Testing Number:</b> 22
<b>Team Member(s) Responsible:</b>	Baanee Singh Josh Subhan
<b>Inputs:</b>	<ul style="list-style-type: none"> <li>- Current documentation</li> <li>- Updated logic and test cases</li> </ul>
<b>Tests:</b>	<ul style="list-style-type: none"> <li>- JavaDocs</li> <li>- In-line comments for file validation logic.</li> <li>- Add multi-file usage example to README.md.</li> </ul>
<b>Outputs:</b>	<ul style="list-style-type: none"> <li>- JavaDocs, README updates.</li> </ul>
<b>Passed or Failed:</b>	Passed
<b>Date:</b>	3/26/2025