

# Sprint Backlog

## COMMITTED BACKLOG ITEMS

<p><b>#1 Algorithms Fully Implemented</b></p> <p>As a User</p> <p>I want all algorithms that determine the results of an election (IRV, OPLV) work as described in the SRS document and SDD, so that I can utilize the software confidently.</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"><li>- Tests are written for all algorithm paths</li><li>- Algorithm produces results for all algorithm expected path results including ties</li></ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"><li>- All IO Exception Handling is implemented</li><li>- Passes regression testing</li><li>- Meets all acceptance criteria</li><li>- In-line documentation and javadocs have been updated.</li></ul> <p><b>Size:</b> Small</p> <p><b>Status:</b> To-do</p>
<p><b>#2 Remove and Store Invalidated Ballots</b></p> <p>As a user of the voting system</p> <p>I want IRV to only use valid ballots (ballots with at least half of the candidates ranked) and to store invalidated ballots into a separate audit file so that the election is fair and only validated ballots are used</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"><li>- Audit file of invalidated ballot is produced</li><li>- Calculate election with validated ballots</li><li>- Invalidated ballots are only in the IRV election</li></ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"><li>- Passes Regression Testing</li><li>- Meeting all Acceptance Criteria</li><li>- In-line documentation and javadocs have been updated.</li></ul> <p><b>Size:</b> Medium</p> <p><b>Status:</b> To-do</p>
<p><b>#3 Election Table</b></p> <p>As an official (of an IRV Election)</p> <p>I want file to use a table showing each round of the IRV so that I can see the number of votes that the candidate added/subtracted for each round</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"><li>- Calculate votes added for each candidate in each round</li><li>- Calculate votes subtracted for each candidate in each round</li><li>- Create GUI/Display of table</li></ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"><li>- Able to show feature in a demo</li><li>- Pass all tests (regression, acceptance, unit, etc)</li><li>- Approved by team</li><li>- Meets all Acceptance Criteria</li><li>- In-line documentation and javadocs have been updated.</li></ul> <p><b>Size:</b> Large</p> <p><b>Status:</b> To-do</p>
<p><b>#4 Command Line Argument Filename</b></p> <p>As a user</p> <p>I want to provide the filename as an argument in the commandline so that I do not have to be prompted and to expedite the process of running the election</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"><li>- Take in the command line argument</li><li>- Save the argument as the file name</li><li>- No longer prompt the user, either in the command line or the GUI, for the file name</li></ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"><li>- Feature passes acceptance criteria</li><li>- Feature can be showed in a demo</li><li>- Approved by the team</li><li>- Passes all required tests</li><li>- In-line documentation and javadocs have been updated.</li></ul> <p><b>Size:</b> Small</p> <p><b>Status:</b> To-do</p>
<p><b>#5 Filename Prompt</b></p> <p>As a user</p> <p>I want to only be prompted for the filename on the command line so I can dictate the election file</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"><li>- OPLV &amp; IRV is properly calculated</li><li>- Audit file can recreate the election</li><li>- election is fair</li><li>- Ensure it is Thoroughly Tested - Forms of Regression Testing</li><li>- Documentation</li></ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"><li>- Functionality is able to be demo-ed</li><li>- All tests have been written and passed</li><li>- A log of potential conflicts with components in the system</li><li>- In-line documentation and javadocs have been updated.</li></ul> <p><b>Size:</b> Small</p> <p><b>Status:</b> Done</p>

## NOT STARTED

Documentation of Exception Handling	Handle File I/O exceptions
Documentation	Integrate removal of ballots to pass name of removed ballots file to GUI and Terminal
Test Logs	
Manually Test GUI Component	Documentation
Test Logs	
Meghann: Finalize GUI design after functionality implemented	Design unit tests election table data and integration
	Automated Testing
Test Logs	Testing

## IN PROGRESS

CHRISTINE Test function which determines if a ballot should be removed	CHRISTINE Test that the file with removed ballot has all removed ballots	CHRISTINE Automated Testing for invalid ballots
JAKE Pass in percentage of ballots in OPLV required as flag		
MEGHANN Implement votes added and subtracted in GUI	MEGHANN Functionality to calculate votes added/subtracted in each round	

## COMPLETED

JAKE Handle all other exceptions caused by Scanner in votingSystemFromFile	JAKE Manual Testing of invalid files	JAKE Delete Audit Files Generated By All Tests
JAKE Test Logs of Exception Handling	JAKE GUI is gracefully exited when exception in Main (with manual tests)	CHRISTINE Handle ParseException By Throwing New Invalid File Exception
CHRISTINE Code logic to remove all invalid ballots from the list of ballots that will be used to calculate the election	CHRISTINE Code function to check if a ballot is valid	CHRISTINE Create invalid audit file containing removed invalid ballots
JAKE Design / Implement GUI Interface	MEGHANN Functionality to zero out votes of eliminated candidate	MEGHANN Fix table to show votes when winner by instant majority
MEGHANN Manually test printing votes for candidates in each round	MEGHANN Initial revision of Table: Candidate Names, # of Votes for each round	
WATERFALL Read in filename command line argument	WATERFALL Documentation	WATERFALL Automated Testing
	WATERFALL Test file name as command line argument	WATERFALL Test Logs
WATERFALL Implement GUI	WATERFALL Documentation	

<p><b>#6 GUI Filename prompt</b> As an election official I want a GUI to use to prompt me for the file name instead of a text prompt so I do not have to use a terminal</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"> <li>- Ensure it is thoroughly tested</li> <li>- Allow user to look for file on their local device</li> <li>- GUI appears when program is run</li> </ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"> <li>- Passes all regression test</li> <li>- Agreed by team</li> <li>- In-line documentation and Javadocs have been updated.</li> <li>- Meets all Acceptance Criteria</li> </ul> <p><b>Size:</b> Large</p> <p><b>Status:</b> Done</p>
<p><b>#7 Handle Upwards of 100,000 Ballots</b> As a user I want the program to be able to handle upwards of 100,000 ballots so that I can run a large election or test the efficiency of the program</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"> <li>- The program can run with an election file that contains 100,000 ballots</li> <li>- The program will output the correct results.</li> <li>- The program still meets the efficiency requirements while running this large number of ballots</li> <li>- Program will still operate and not break because of this</li> </ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"> <li>- All tests are passed</li> <li>- Feature can be showed in a demo</li> <li>- Acceptance criteria are all met</li> <li>- Accepted by the team</li> <li>- In-line documentation and Javadocs have been updated.</li> </ul> <p><b>Size:</b> Medium</p> <p><b>Status:</b> Done</p>
<p><b>#8 Print Summary Report</b> As an election official I want the audit report and a short report containing data, type of election, candidates, the number of seats, and the winner(s) of the election so that I can have a printable version to give to the election certification officials</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"> <li>- Printable report</li> <li>- Only contains date, type of election, candidates, the number of seats, and winner(s) of election</li> <li>- Program produces both audit report and short report</li> </ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"> <li>- Passes all regression test</li> <li>- Agreed by team</li> <li>- Passes Acceptance Criteria</li> </ul> <p><b>Size:</b> Medium</p> <p><b>Status:</b> To-do</p>
<p><b>#9 Add Votes Dynamically To Program Using GUI</b> As an election official using the GUI, I want a way to add votes dynamically to the program so that I can send the information directly to the system without the use of a file.</p> <p><b>Acceptance Criteria:</b></p> <ul style="list-style-type: none"> <li>- Run an election without a file</li> <li>- A user can input details about the election that would normally be retrieved from the file (e.g. number of candidates)</li> <li>- Ballots are dynamically added to the election</li> <li>- Program will account for those and output the correct results</li> <li>- Tests are written in JUnit for all methods and classes created during the implementation of this feature</li> </ul> <p><b>Definition of Done:</b></p> <ul style="list-style-type: none"> <li>- Feature passes all tests</li> <li>- Feature can be showed in a demo</li> <li>- Accepted by the team</li> <li>- Well documented in Javadocs and inline</li> <li>- Meets Acceptance Criteria</li> </ul> <p><b>Size:</b> Medium</p> <p><b>Status:</b> To-do</p>

