Sprint Backlog

COMMITTED BACKLOG ITEMS

#1 Algorithms Fully Implemented
As a User
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.

Acceptance Criteria:
- Tests are Written for all Algorithm paths
- Algorithm produces results for all algorithm expected path results including ties

Definition of Done:
- All IO Exception Handling is implemented
- Passes regression testing
- Meets all acceptance criteria
- In-line documentation and Javadocs have been updated.

Size: Small

Status: To-do

#2 Remove and Store Invalidated Ballots
As a user of the voting system,
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

Acceptance Criteria:
- Audit file of invalidated ballot is produced
- Calculate election with validated ballots
- Invalidated ballots are only in the IRV election

Definition of Done:
- Passes Regression Testing
- Meeting all Acceptance Criteria
- In-line documentation and Javadocs have been updated.

Size: Medium

Status: To-do

#3 Election Table
As an official (of an IRV Election)
I would like to see 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

Acceptance Criteria:
- Calculate votes added for each candidate in each round
- Calculate votes subtracted for each candidate in each round
- Create GUI/Display of table

Definition of Done:
- Able to show feature in a demo
- Pass all tests (regression, acceptance, unit, etc)
- Approved by team
- Meets all Acceptance Criteria
- In-line documentation and Javadocs have been updated.

Size: Large

Status: To-do

# 4 Command Line Argument Filename
As a user
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

Acceptance Criteria:
- Take in the command line argument
- Save the argument as the file name
- No longer prompt the user, either in the command line or the GUI, for the file name

Definition of Done:
- Feature passes acceptance criteria
- Feature can be showed in a demo
- Approved by the team
- Passes all required tests
- In-line documentation and Javadocs have been updated.

Size: Small

Status: To-do

NOT STARTED	IN PROGRESSS	COMPLETED
Documentation of Exception Handling Handle File I/O exceptions		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
Documentation Integrate removal of ballots to pass name of removed ballots file to GUI and Terminal Test Logs	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	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
Manually Test GUI Component Documentation Test Logs Meghann: Finalize GUI design after functionality implemented Design unit tests election table data and integration Automated Testing	MEGHANN Implement votes added and subtracted in GUI MEGHANN Functionality to calculate votes added/subtracted in each round	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

#5 Filename Prompt As a user I want to only be prompted for the filename on the command line so I can dictate the election file **Acceptance Criteria:** - OPLV & IRV is properly calculated - Audit file can recreate the election - election is fair - Ensure it is thoroughly Tested - Forms of Regression Testing - Documentation **Definition of Done:** - Functionality is able to be demo-ed - All tests have been written and passed - A log of potential conflicts with components in the system - In-line documentation and Javadocs have been updated. **Size:** Small **Status**: Done			WATERFALL Implement GUI WATERFALL Documentation WATERFALL Testing WATERFALL Test Logs
#6 GUI Filename prompt As a electoral 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 **Acceptance Criteria:** - Ensure it is throughly tested - Allow user to look for file on their local device - GUI appears when program is run **Definition of Done:** - Passes all regression test - Approved by team - In-line documentation and Javadocs have been updated. - Meets all Acceptance Criteria **Size:** Large **Status**: Done			WATERFALL Implement Error Popups for Invalid File Name WATERFALL Implement GUI WATERFALL Implement Search Files on Machine WATERFALL Implement Flag for NO GUI WATERFALL Documentation for invalid file name popup
#7 Handle Upwards of 100,000 Ballots 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 **Acceptance Criteria:** - The program can run with an election file that contains 100,000 ballots - The program will output the correct results - The program still meets the efficiency requirements while running this large number of ballots - Program will still operate and not break because of this **Definition of Done:** - All tests are passed - Feature can be showed in a demo - Acceptance criteria are all met - Accepted by the team - In-line documentation and Javadocs have been updated. **Size:** Medium **Status:** Done			WATERFALL Create File with 100,000 ballots WATERFALL Write Efficiency Test for 100,000 Votes WATERFALL Documentation WATERFALL Test Logs
#8 Print Summary Report As an election official I want the audit report and a short report containing date, 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 **Acceptance Criteria:** - Printable report - Only contains date, type of election, candidates, the number of seats, and winner(s) of election - Program produces both audit report and short report **Definition of Done:** - Passes all regression test - Approved by team - Able to show feature in demo - Passes Acceptance Criteria **Size:** Medium **Status**: To-do		CASSANDRA Documentation for stringbuilder of print summary CASSANDRA Test Logs for GUI Component	JAKE Design / Implement GUI Interface CASSANDRA Code stringbuilder to collect quick print summary CASSANDRA Manually Test print summary JAKE Documentation for GUI
#9 Add Votes Dynamically To Program Using GUI 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. **Acceptance Criteria:** - Run an election without a file - A user can input details about the election that would normally be retrieved from the file (e.g. number of candidates). - Ballots are dynamically added to the election - Program will account for these and output the correct results - Tests are written in JUNIT for all methods and classes created during the implementation of this feature **Definition of Done:** - Feature passes all tests - Feature can be showed in a demo - Accepted by the team - Well documented in javadocs and inline - Meets Acceptance Criteria **Size**: Medium **Status**: To-do	Design GUI Interface for adding votes Implement GUI Functions for adding votes Manually test dynamically input votes Documentation Automated Testing Test Logs		JAKE Design / Implement GUI menu bar