

| Sprint Backlog 4/22/2022 | | | | | | | | | | | | |
|--------------------------|--|-------------------|---|------------------|---------|---------------------|---|--|-----------|--|--|--|
| PBI ID | User Story | Owner | Tasks | Est Time (hours) | Status | Who's working on it | Acceptance Criteria | Definition of Done | Due Date | | | |
| #1 | As an election official, I want to be able to determine a number of winners via IRV so that I don't have to calculate the winners myself | Election Official | Fix file parsing to allow multiple winners ensure that file parsing works properly | 0.5 | Done | | The user can submit a file for input | 1) All previous regression tests pass 2) Regression tests for new functionality run automatically every build 3) Code written in pairs/ viewed by other members 4) Duplicate code is removed through refactoring 5) Messy code cleaned up through refactoring 6) manual , exploratory testing has been conducted 7) Checkout and build fully reproducible (1-2 commands) | 4/20/2022 | | | |
| | | | Fix calcResults() function to allow multiple winners | 0.5 | Done | | command can be run to calculate correct IRV election results | | 4/20/2022 | | | |
| | | | Ensure that correct winners can be retrieved by user | 0.5 | Done | | receives correct results of the election | | 4/20/2022 | | | |
| | | | Unit testing | 2 | Started | Ann, Sean | all tests pass and system is able to test runtime of 100000 ballots | | 4/20/2022 | | | |
| | | | bonus: split calcResults() into readinput() and calcResults() (low priority) | 0.5 | Done | | Functionality of calcResults is maintained, but all functionality that involves reading from files is moved to readinput() | | 4/20/2022 | | | |
| | | | | | | | | | | | | |
| #2 | As an election official, I want to determine a number of winners via OPL so that I don't have to calculate the winners myself | Election Official | Ensure that file parsing works properly | 0.5 | Done | | The user can submit a file for input | 1) All previous regression tests pass 2) Regression tests for new functionality run automatically every build 3) Code written in pairs/ viewed by other members 4) Duplicate code is removed through refactoring 5) Messy code cleaned up through refactoring 6) manual , exploratory testing has been conducted 7) Checkout and build fully reproducible (1-2 commands) | 4/20/2022 | | | |
| | | | Fix calcResults() function to calculate the winners correctly | 1.5 | Done | | command can be run to calculate correct OPL election results | | 4/20/2022 | | | |
| | | | Ensure that correct winners can be retrieved by user | 0.5 | Started | Ann, Sean | receives correct results of the election | | 4/20/2022 | | | |
| | | | Unit Testing | 3 | Started | Ann, Sean | all tests pass and system is able to test runtime of 100000 ballots | | 4/20/2022 | | | |
| | | | bonus: split calcResults() into readinput() and calcResults() (low priority) | 0.5 | Done | | Functionality of calcResults is maintained, but all functionality that involves reading from files is moved to readinput() | | 4/20/2022 | | | |
| | | | | | | | | | | | | |
| #7 | As an election official, I want to be able to bring in multiple files instead of a single file | Election Official | Change input loop so that "filename" always adds the given filename to a list of files, rather than overwriting previous filenames | 0.25 | Done | | "filename" command does not cause previous input filenames to be overwritten, and instead adds the supplied filename to a list of filenames | 2) Regression tests for new functionality run automatically every build 3) Code written in pairs/ viewed by other members 4) Duplicate code is removed through refactoring 5) Messy code cleaned up through refactoring 6) manual , exploratory testing has been conducted 7) Checkout and build fully reproducible (1-2 commands) | 4/22/2022 | | | |
| | | | give the filehandler a list of filenames rather than just one filename | 0.25 | Done | Shey, Jasper | FileHandler no longer keeps track of a single filename. Instead, FileHandler maintains a list of filenames. | | 4/22/2022 | | | |
| | | | replace FileHandler.setinputFile() with something like FileHandler.addinputFile(), this new method should behave similarly to setinputFile but should add valid files to the list of filenames rather than just overwriting the previous filename | 0.25 | Done | Shey, Jasper | FileHandler.setinputFile() no longer exists, and FileHandler.addinputFile() is created with full functionality. New function checks the supplied filename for if it exists. If it exists, then it is added to the list of filenames. Otherwise it is ignored. | | 4/22/2022 | | | |
| | | | possibly change from creating a FileHandler when the first file is passed in, to creating one when the program starts? | 0.25 | Done | Shey, Jasper | System creates a FileHandler object when the program starts up, this object is closed when the program is closed via the "exit" command. The System no longer creates a FileHandler object when the first valid filename is passed in. | | 4/22/2022 | | | |
| | | | create a getter for FileHandler, filenames.size (probably useful for adding multiple file support to electionOPL.electionIRV) | 0.25 | Done | Shey, Jasper | FileHandler has a getter function which returns the FileHandler's list of files. | | 4/22/2022 | | | |
| | | | update electionOPL to support reading input from multiple files | 0.5 | Done | Shey, Jasper | an OPL election can be ran which draws its inputs from more than one input file | | 4/22/2022 | | | |
| | | | update electionIRV to support reading input from multiple files | 0.5 | Done | Shey, Jasper | an IRV election can be ran which draws its inputs from more than one input file | | 4/22/2022 | | | |
| | | | possibly create function which makes it possible to remove files from the list of input files (low priority) | 0.5 | Started | Shey, Jasper | the user can remove a filename from the list of filenames via a command. | | 4/22/2022 | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |

| | | | | | | | | |
|----|---|--|--|------|-------------|-----------|--|-----------|
| | | | Unit testing | 2 | Not Started | Ann. Sean | All unit tests for each of the tasks for this PJ1 pass, along with all previous unit tests. | 4/22/2022 |
| | | | change the code which reads the ballots in so that it checks each ballot for if it is valid. Invalid ballots will be disregarded/added to audit file | | Not Started | | An IRV election can be fed invalid ballots via input file(s) and the election will automatically ignore and mark the invalid ballots in the relevant audit file | |
| #9 | As an election official, I want to be able to automatically filter out the ballots which have less than half the candidates ranked in an IRV election | | update FileHandler to support reading an audit file (add a function) | 0.5 | Not Started | | 1) All previous regression tests pass 2) Regression tests for new functionality run automatically every build 3) Code written in pairs/ viewed by other members 4) Duplicate code is removed through refactoring 5) Messy code cleaned up through refactoring 6) manual , exploratory testing has been conducted 7) Checkout and build fully reproducible (1-2 commands) | 4/25/2022 |
| | | | Unit testing | 0.5 | Not Started | | all tests pass and system is able to test runtime of 100000 ballots | 4/25/2022 |
| | | | create ElectionPO class | 0.25 | Not Started | | ElectionPO class exists and extends the Election class | 4/27/2022 |
| #4 | As an election official, I want to be able to bring in PO election ballots via a .csv file, so that I don't have to input the individual ballots manually | | write ElectionPO.readInputs() which reads all supplied files for inputs. | 1.5 | Not Started | | 1) All previous regression tests pass 2) Regression tests for new functionality run automatically every build 3) Code written in pairs/ viewed by other members 4) Duplicate code is removed through refactoring 5) Messy code cleaned up through refactoring 6) manual , exploratory testing has been conducted 7) Checkout and build fully reproducible (1-2 commands) | 4/27/2022 |
| | | | If the default Candidate class is not enough to complete this, then write a CandidatePO class | 0.5 | Not Started | | CandidatePO class exists and has all needed functionality which the Candidate class did not have. | 4/27/2022 |
| | | | unit testing | 2 | Not Started | | all tests pass and system is able to test runtime of 100000 ballots | 4/27/2022 |
| | | | create ElectionPO.calcResults() which behaves similarly to ElectionIRV.calcResults() and ElectionOPL.calcResults() | 1.5 | Not Started | | command can be run to calculate correct PO election results | 4/27/2022 |
| #3 | As an election official, i want to be able to determine a single winner via PO (the candidate who received the most ballots) | | ensure that correct winners can be retrieved by user | 0.5 | Not Started | | 1) All previous regression tests pass 2) Regression tests for new functionality run automatically every build | 4/27/2022 |
| | | | create input command to request PO election | 0.5 | Not Started | | existing input command can read a PO election option | 4/27/2022 |
| | | | unit testing | 2 | Not Started | | all tests pass and system is able to test runtime of 100000 ballots | 4/27/2022 |
| | | | write a fair random coin flip (possibly use the one which was already written) | 0.5 | Not Started | | Coin flip in case of tie is fair and randomized | 4/27/2022 |