	Due Date	4/20/2022	4/20/2022	4/20/2022	4/20/2022	4/20/2022	4/20/2022	4/20/2022	4/20/2022	4/20/2022	4/20/2022	4/22/2022	4/22/2022	4/22/2022	4/22/2022	4/22/2022	4/22/2022	4/22/2022	4/22/2022	
	Definition of Done	1) All previous regression tests pass 3) Cade written for new functionality run automatically 3) Code written in pairs' viewed by other members 4) Duplicates does it seman-ved through reflectoring 5) Messy code cleaned up through reflectoring 6) Messy code cleaned up through reflectoring 7) Checkout and build fully reproducible (1-2 commands)					1) All previous regression tests pass 3) Cade writen stornew functionality run automatically 3) Code writen in pairs' viewed by other members 4) Duplicates aced is removed through reflacting 5) Messy code cleaned up through reflacting 6) Messy code cleaned up through reflacting 6) manual. exploratory resting has been conducted 7) Checkout and build fully reproducible (1-2 commands)				7) Checkout and build fully reproducible (1-2 commands)	1) All previous regression tests pass sometimes for new functionality run automatically easy build regression tests for new functionality to automatically sometimes to the regression tests to the members of Duplicate code is removed through refactoring 6) menual respondency testing has been condicioned for menual respondency testing has been conducted 7). Checkout and build fully reproducible (1-2 commands).								
Sprint Backlog 4/21/2022	Acceptance Criteria Definition of Done	The user can submit a file for input	command can be run to calculate correct IRV election results	receives correct results of the election	all tests pass and system is able to test runtime of 100000 ballots	Functionality of calcResults is maintained, but all functionality that involves reading from files is moved to readinput()	The user can submit a file for input	command can be run to calculate correct OPL election results	receives correct results of the election	all tests pass and system is able to test runtime of 100000 ballots	Functionality of calcResults is maintained, but all functionality that involves reading from files is moved to readinput()	"flename" command does not cause previous input flesnames to be overwritten, and instead adds the supplied flename to a list of flenames	FileHandler no longer keeps track of a single filename. Instead, FileHandler maintains a list of filenames.	FileHandler.setinputfile() no briger exists, and FileHandler.setinputfile() is created with full functionality. New function checks the supplied filenam of rif (Lessis, If it exists, then it is added to the list of filenames. Otherwise it is grored.	System creates a Fletander object when the program starts up, this object is cosed when the program is chosed when the program is chosed with the "exif" command. The System no binger creates a Fletander object when the first vidil filename is passed in "life in the life in the start of the man is passed in "life in the life."	FileHandler has a getter function which returns the FileHandler's list of files.	an OPL election can be ran which draws its inputs from more than one input file	an IRV election can be ran which draws its inputs from more than one input file	the user can remove a filename from the list of filenames via a command.	
	Who's working on it				Ann, Sean				Ann, Sean	Ann, Sean		Shev, Jasper	Shey, Jasper	Shev, Jasher	New Jasener	Shey, Jasper	Shev, Jasper	Shey, Jasper	Shey, Jasper	
	Status	Done	Done	5 Done	Started 2	Done	Done	Done	0.5 Started	Started 3	Done	Not Started	Not Started	Not Started	Not Started	Not Started	Not Started	Not Started	Not Started	
	Est Time (hours)	0.5	0.5	0.5		0.5	0.5	1.5	70		0.5	0.25	0.25		0.25	0.25	0.5	0.5	0.5	
		Fix file parsing to allow multiple winners/ensure that file parsing works properly	Fix calcResults() function to allow multiple winners	Ensure that correct winners can be retrieved by user	Unit testing	bonus: split calcResults() into readinput() and calcResults() (low priority)	Ensure that file parsing works properly	Fix calcResults() function to calculate the winners correctly	Ensure that correct winners can be retrieved by user	Unit Testing	bonus: split calcResults() into readinput() and calcResults() (low priority)	Change input loop so that "flename" always adds the given filename to a list of files, rather than overwriting previous flenames	give the filehandler a list of filenames rather than just one filename	replace Flerlander.sethputFle() with something like Flerlander. addition_fle(), this new method should be have similarly to sethputFle turn stould add valid files to the last of filenames rather the bust overwriting the previous filename.	possibly change from creating a FileHandler when the first file is seased in, to creating one when the procram starts?	create a getter for FlleHandler. filenames (probably useful for adding multiple file support to electionOPL/electionIRV)	update electionOPL to support reading input from multiple files	update electionIRV to support reading input from multiple files	possibly create function which makes it possible to remove files from the list of input files (low priority)	
	Owner	Official									928	Bection Official								
	User Story (	n official I able to rumber of auchies to rumber of a value in a don't have a don't have a fer in me winners via a don't have a fer in a value										As an election official. I want to be able to bring in multiple files instead of a single file								
	PBIID	#								#5		L			44.					

4/22/2022	4/25/2022	4/25/2022	4/25/2022	4/27/2022	4/27/2022	4/27/2022	4/27/2022	ı	4/27/2022	4/27/2022	4/27/2022	4/27/2022	4/27/2022		
	1) All previous regression tests pass 2) Recression tests for new functionality run automatically	3) Code written in parist viewed by other members 4) Duplicate code is removed through refactoring 5) Messy code desended up fruoding refactoring 6) Messy code desended up fruoding refactoring 6) manual. exploratory testing has been conducted 7) Checkout and build fully reproducible (1-2 commands)			1) All previous regression tests pass 2) Regression tests for new functionality run automatically every build	1) All previous regression tests pass 2) Regression tests for new functionality run automatically 3) Code written in parts/ viewed by other members 4) Duplicate code is removed fincular freatoning 4) Nessy code cleaned up through refaccining 6) Messy code cleaned up through refaccining 6) manual exploratory testing has been conducted 7) Checkout and build fully reproducible (1-2 commands) 4									
All unit tests for each of the tasks for this PBI pass, along with all previous unit tests.	An IRV election can be fed invalid balloss via input file(s) and the election will automatically ignore and mark the invalid ballots in the relection will he release and such that the release and th	This	all tests pass and system is able to test runtime of 100000 ballots	ElectionPO class exists and extends the Election class	ElectionPO.readInputs() exists and can read all candidates/votes in from all supplied input files.	CandidatePO class exists and has all needed functionality which the Candidate class did not have.	all tests pass and system is able to test runtime of 100000 ballots	command can be run to	calculate correct PO election results	receives correct results of the election	existing input command can read a PO election option	all tests pass and system is able to test runtime of 100000 ballots	Coin filp in case of tie is fair and randomized		
Ann, Sean															
Not Started	Not Started	Not Started	Not Started	Not Started	Not Started	Not Started	Not Started		Not Started	itarted	Not Started	Not Started	Not Started		
Not S	o A		Note	Not 8	Not 9	Not 8	Not 8	ľ	Not 5	0.5 Not Started	Not 5	Not 8	Not 8		
Unit testing	change the code which reads the ballos in so that it checks each ballot for if it is valid. Invalid to hanks will be disregarded/added to another the conditions.	update FileHandler to support this new audit file (add a function)	Unit testing	create ElectionPO dass	write ElectionPO.readInputs() which reads all supplied files for inputs.	If the default Candidate class is not enough to complete this, then write a CandidatePO class	unit testing	create ElectionPO.calcResults()	ElectionIRV.calcResults() and ElectionOPL.calcResults()	ensure that correct winners can be retrieved by user	create input command to request PO election	unit testing	write a fair random coin flip (possibly use the one which was already written)		
	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			As an election official I want to be able to thing in PO election ballots via a csr file as that I don't have to input the individual ballots manually					As an election official, i went to be able to determine a single winer to be able to determine as not be able to the candidate who received the most ballots)					
		6#			4.4					Ę					