To					
Parent PBI ID	**Parent PBI Description**	**Task ID**	**Task Description**	**Owner(s)**	**Task Status**
1	Run an algorithm to determine the results of a Plurality type election	1	Refactor code for plurality elections where possible. Make sure code is efficient and well commented.	Amir	Done
1	Run an algorithm to determine the results of a Plurality type election	2	Test: The program runs without crashing and computes correct results for an election with 0 ballots.	Arun	Done
1	Run an algorithm to determine the results of a Plurality type election	3	Test: The program runs without crashing and computes correct results for an election with 100,000 ballots.	Arun	Done
1	Run an algorithm to determine the results of a Plurality type election	4	Test: The program runs without crashing and computes correct results for an election with 1 candidate	Arun	Done
1	Run an algorithm to determine the results of a Plurality type election	5	Test: The program runs without crashing and computes correct results for an election with 10 candidates	Arun	Done
1	Run an algorithm to determine the results of a Plurality type election	6	Test: The program runs without crashing and computes correct results for an election that contains only ballots that do not rank candidates	Arun	Done
1	Run an algorithm to determine the results of a Plurality type election	7	Test: The program runs without crashing and computes correct results for an election with 10 candidates	Arun	Done
1	Run an algorithm to determine the results of a Plurality type election	8	Test: The program runs without crashing and computes correct results for an election with a two-way tie	Arun	Done
1	Run an algorithm to determine the results of a Plurality type election	9	Test: The program runs without crashing and computes correct results for an election with a four-way tie	Arun	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	10	Test: The program runs without crashing and computes correct results for an election with 0 ballots.	Taylor	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	11	Test: The program runs without crashing and computes correct results for an election with 100,000 ballots.	Taylor	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	12	Test: The program runs without crashing and computes correct results for an election with 1 candidate	Taylor	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	13	Test: The program runs without crashing and computes correct results for an election with 10 candidates	Taylor	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	14	Test: The program runs without crashing and computes correct results for an election that contains only ballots that do not rank candidates	Taylor	Done

2	Run an algorithm to determine the results of a STV with Droop Quota type election	15	Test: The program runs without crashing and computes correct results for an election with 5 candidates, 15 ballots, and 1 winner	Taylor	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	16	Test: The program runs without crashing and computes correct results for an election with 5 candidates, 15 ballots, and 2 winners	Taylor	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	17	Test: The program runs without crashing and computes correct results for an election with 5 candidates, 15 ballots, and 3 winners	Taylor	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	18	Test: The program runs without crashing and computes correct results for an election with 5 candidates, 15 ballots, and 4 winners	Taylor	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	19	Test: The program runs without crashing and computes correct results for an election with 5 candidates, 15 ballots, and 5 winners	Taylor	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	20	Test: Write unit tests to make sure the candidate class works correctly.	Arun	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	21	Add common dependencies for tests.	Taylor	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	22	Change existing tests to use the common dependencies.	Taylor	Done
2	Run an algorithm to determine the results of a STV with Droop Quota type election	23	Create insructions to compile and run the program and tests.	Taylor	Done
3	Have the ballot files loaded shuffled	24	Test: Check if the first ballot id does not match with the given ballot id while shuffling.	Arun	Done
4	Have the program check if any of the ballots added are invalid	25	Planning and Design: Look up and determine the best approach to preprocess the invalidated ballots and discard them	Amir	Done
4	Have the program check if any of the ballots added are invalid	26	Write code to identify invalid ballots for elections and discard them before running the election.	Amir	Done
4	Have the program check if any of the ballots added are invalid	27	Document the code to identify invalid ballots. Also change documentation for existing code that was changed (if any) by this.	Amir	Done
4	Have the program check if any of the ballots added are invalid	28	Test: Check if the program correctly identifies the invalidated ballots passed in and discards them before running the election.	Arun	Done
5	Have the program report the invalidated ballots in a Droop election by saving them to disk in a file.	29	Write code to save the invalidated ballot's information to a file, then to save that file to disk.	Amir	Done
5	Have the program report the invalidated ballots in a Droop election by saving them to disk in a file.	30	Document the code to save invalid ballots.	Amir	Done
5	Have the program report the invalidated ballots in a Droop election by saving them to disk in a file.	31	Test: Check if the invalidated report file is created and written in the disk with the expected contents.	Arun	Done

6	Select multiple ballot files with a GUI.	32	Write code to provide the user with a file chooser that allows them to select only ballot files. The user should be able to select multiple files.		Done
6	Select multiple ballot files with a GUI.	33	Document the code for the file chooser for multiple ballot files.		Done
6	Select multiple ballot files with a GUI.	34	Test: Verify that the system correctly gathers all ballots and candiates for multiple ballot files with 5 candidates and 30 ballots.	Nicholas	Done
7	Search for ballot files on the disk with a GUI.	35	Ensure that the file chooser provides search functionality for the user.	Nicholas	Done
7	Search for ballot files on the disk with a GUI.	36	Test: Verify that the explorer opens and can find ballot files successfully.	Nicholas	Done
8	Select a directory to load ballot files from using a GUI.	37	Write code to provide the user with a file chooser that allows them to select only directories. The user should be able to select only one directory.		Done
8	Select a directory to load ballot files from using a GUI.	38	Document the code for the file chooser for a single directory.		Done
8	Select a directory to load ballot files from using a GUI.	39	Test: Verify that the system correctly gathers all ballots and candiates for single/multiple ballot file(s) with 5 candidates and 30 ballots. Subdirectories should not be included, only the directory selected.	Nicholas	Done
9	Create a short report of election results and save it to disk as a file.	40	Create a method to interpret the results of an election and produce the info needed for the short report as a String.	Amir	Done
9	Create a short report of election results and save it to disk as a file.	41	Write code to save this String to a file as the short report.	Amir	Done
9	Create a short report of election results and save it to disk as a file.	42	Test: Verify that the report generated contains the correct election results	Amir	Done
9	Create a short report of election results and save it to disk as a file.	43	Test: Verify that the file is saved to disk in the expected location with the expected name	Amir	Done
9	Create a short report of election results and save it to disk as a file.	44	Document the new code for creating the short report.	Amir	Done
6	Select multiple ballot files with a GUI.	45	Test: Check that null is returned if the user quits the file selection process.	Nicholas	Done
6	Select multiple ballot files with a GUI.	46	Test: Check that the user can select a single file.	Nicholas	Done
6	Select multiple ballot files with a GUI.	47	Test: Check that the user can select a few specific files out of a larger group.	Nicholas	Done
6	Select multiple ballot files with a GUI.	48	Test: Check that the user can select all the files in a directory.	Nicholas	Done
8	Select a directory to load ballot files from using a GUI.	49	Test: Check that null is returned if the user quits the file selection process.	Nicholas	Done

2	Run an algorithm to determine the results of a Plurality type election	50	Review the comments / documentation for Javadocs for the STVDroopController class.	Amir	Done
1	Run an algorithm to determine the results of a Plurality type election	51	Review the comments / documentation for Javadocs for the PluralityController class.	Amir	Done
1	Run an algorithm to determine the results of a Plurality type election	52	Review the comments / documentation for Javadocs for the ElectionController class.	Amir	Done
1	Run an algorithm to determine the results of a Plurality type election	53	Review the comments / documentation for Javadocs for the BallotFactory class.	Nicholas	Done
1	Run an algorithm to determine the results of a Plurality type election	54	Review the formatting of code / comments / file header for the Ballot class.	Nicholas	Done
1	Run an algorithm to determine the results of a Plurality type election	55	Review the formatting of code / comments / file header for the BallotFactory class.	Nicholas	Done
1	Run an algorithm to determine the results of a Plurality type election	56	Review the formatting of code / comments / file header for the BallotFactoryTest class.	Nicholas	Done
1	Run an algorithm to determine the results of a Plurality type election	57	Review the formatting of code / comments / file header for the BallotFileChooser class.	Nicholas	Done
1	Run an algorithm to determine the results of a Plurality type election	58	Review the formatting of code / comments / file header for the ElectionController class.	Amir	Done
1	Run an algorithm to determine the results of a Plurality type election	59	Review the formatting of code / comments / file header for the ElectionControllerTest class.	Amir	Done
1	Run an algorithm to determine the results of a Plurality type election	60	Review the formatting of code / comments / file header for the HelpWindow class.	Nicholas	Done
2	Run an algorithm to determine the results of a Plurality type election	61	Review the formatting of code / comments / file header for the InvalidationTest class.	Amir	Done
1	Run an algorithm to determine the results of a Plurality type election	62	Review the formatting of code / comments / file header for the MainWindow class.	Nicholas	Done
1	Run an algorithm to determine the results of a Plurality type election	63	Review the formatting of code / comments / file header for the PluralityController class.	Arun	Done
1	Run an algorithm to determine the results of a Plurality type election	64	Review the formatting of code / comments / file header for the PluralityControllerTest class.	Arun	Done
2	Run an algorithm to determine the results of a Plurality type election	65	Review the formatting of code / comments / file header for the STVDroopController class.	Nicholas	Done
2	Run an algorithm to determine the results of a Plurality type election	66	Review the formatting of code / comments / file header for the STVDroopControllerTest class.	Nicholas	Done