

Rank	Product Backlog Item Description	Acceptance Criteria	Effort	Unfinished Tasks	Finished Tasks
1	As the product owner, I would need the OPL and IR algorithms to work fully as described in project 1 so that correct results are being generated.	1. All system and unit tests are passed except for minor bugs like exception handling. 2. Results of both elections are displayed to the screen and are correct. 3. Process of election is written to an audit file and is correct.	Small	1. Check tests 2. See response of graders	
2	As an election official, I need all ballots in an IR election that do not have at least half the candidates ranked to be invalidated and written to file named invalidated_dateofelection_xxx in order to audit the election and because ballots need to have at least half the candidates be ranked to be valid and impact the election.	1. All system and unit tests are passed except for minor bugs like exception handling. 2. File invalidated_dateofelection_xxx is written to the directory. 3. It can be shown in the file that removed ballots are the correct ones and are all the invalid ballots in demos and testing.	Medium	1. Code for checking validation of ballots 2. Code for writing the file 3. Code for testing 4. Verify correctness of election	
3	As an election official, I would like to see a table of the election showing each round of the IRV and the number of votes that the candidate added/subtracted for that round. I would like the table should be displayed to the screen so that I can inspect the correctness of the runoffs and understand it easily.	1. All system and unit tests are passed except for minor bugs like exception handling. 2. Table is formatted properly and displayed to the official after the election is done. 3. The number of votes distributed for each candidate after each election is correct.	Large	1. Verify current IRV is correct. 2. Code for managing candidate vote counts 3. Code for tracking the addition and subtraction 4. Code for laying out information in the system 5. Code for displaying the table 6. Code for testing	

4	I, as the user, would like two ways to run the election file. One would to use a command line argument and the other, a prompt for the name of the file. This is the only input that I would like to be prompted for and only if I do not run a command line argument.	1. All system and unit tests are passed except for minor bugs like exception handling. 2. Both methods work and produce the same results. 3. Only one of the options is working at one time.	Small	1. Code for command line 2. Code for passing in file name 3. Unit and system test code	
5	I as the user, would like the filename to be the only information requested from me because inputting other information is tedious and prone to error.	1. All system and unit tests are passed except for minor bugs like exception handling. 2. Prompt only asks user for filename. 3. Election can be done by reading information from the file itself.	Small	1. Code for requesting file name 2. Test code	
6	I, as the user, would like to be prompted for the filename via a GUI instead of a text prompt and I would like window to appear and be able to type in a file name or look for a file on disk so that I am not working in the terminal as the official.	1. All system and unit tests are passed except for minor bugs like exception handling. 2. GUI is shown to the user. 3. The correct file is read and the correct results are generated.	Large	1. Code for GUI window formatting 2. Code for prompting file name 3. Code for accepting and reading in file name from the GUI 4. Code to allow file searching on disk 5. Code for system testing and unit testing	