

| Committed PBIs | Not started | In progress | Completed |
|----------------|---|-------------|--|
| User story 1 | | | - Reuse the source code from the first project |
| User story 2 | <ul style="list-style-type: none"> - Erase the code related to shuffling option - Call shuffling at the beginning of the vote counting process | | |
| User story 3 | <ul style="list-style-type: none"> - Redesign the interface to take only a file name - Redo the parseInput to read a new format of the input file | | |
| User story 4 | <ul style="list-style-type: none"> - Redesign the interface to read the file name from the command line - Redo the parseInput | | |
| User story 6 | <ul style="list-style-type: none"> - add the new function checkBallot in the Election class to check for the validity of a ballot - Modify the format of the audit file to store a list of invalidated ballots | | |
| User story 9 | <ul style="list-style-type: none"> - add the new function generateReport in the Election class to generate a short report of the election - use the generateReport function appropriately in main function - generate a message of success/failure after generating a report | | |

| | |
|---------------------|--|
| User story 1 | As the product owner, I want my program to be able to run the droop and plurality algorithms to ensure the equity of the election. |
| Acceptance criteria | <ol style="list-style-type: none"> 1. Accept only the CSV file generated from Linux machine 2. Validate information in the .csv file 3. Generate a message of success/failure 4. Store the result to the output file 5. Store the vote counting process to the audit file |
| Effort | medium |

| | |
|---------------------|---|
| User story 2 | As an election official, I do not want the program to ask about shuffle or no shuffle because the program should be able to shuffle without user having to turn on and off the shuffle |
| Acceptance criteria | <ol style="list-style-type: none"> 1. Consistently give the correct result of the election 2. The program does not ask users for shuffling option 3. Program should set shuffle to default |
| Effort | Small |

| | |
|---------------------|--|
| User story 3 | As an officer, I want the new interface that accepts only a file name of csv file so that I don't have to type in all information. |
| Acceptance criteria | <ol style="list-style-type: none"> 1. The program ran with the correct result as the old interface 2. Validate the input file name 3. Generate a message of success/failure |
| Effort | small |

| | |
|---------------------|--|
| User story 4 | As an election officer, I want to have an ability to run the program using command line argument in case I don't want to use the interface. |
| Acceptance criteria | <ol style="list-style-type: none"> 1. The program ran with the correct result 2. Validate the input file name 3. Generate a success/failure message |
| Effort | small |

| | |
|---------------------|--|
| User story 6 | As an election officer, I want the program to be able to process invalidated ballots so that I don't have to do it at the time of the ballot collection. |
| Acceptance criteria | <ol style="list-style-type: none"> 1. Store a list of invalidated ballots in an audit file 2. Ensure that invalidated ballots do not get distributed 3. The program does not break with invalidated ballots 4. Validate that the ballot has at least half of the candidates ranked |
| Effort | medium |

| | |
|---------------------|--|
| User story 9 | As an election official, I want the program to generate a short report which contains the date, type of election (i.e. droop or plurality), the candidates, the number of seats, and the winners of the election so that I can give it to the election certification officials |
| Acceptance criteria | <ol style="list-style-type: none"> 1. Generate a message of success/failure of creating a report file 2. A report with all required information is generated at the end 3. A report has an extension of .txt |
| Effort | large |