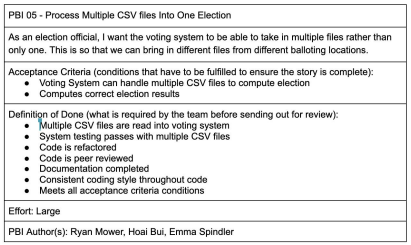
Team 11:

1. Ryan Mower (mower023)
2. Hoai Bui (bui00015)
3. Emma Spindler (spind038)
4. Eric Palmer (palme885)

Sprint Product Backlog

Committed Backlog Items	Not Started	In progress	Complete
PBI 05.) Process Multiple CSV files Into One Election 	Task 1 (Write Code) : Write code to handle multiple CSV files for different election types. --- Task 2 (Peer Review Implemented Code) : Have a peer look over the implemented code that handles multiple CSV election files. --- Task 3 (Refactor Code) : Go though code and refactor it, ensuring it is readable and understandable. --- Task 4 (Ensure Consistent Coding Style) : Go though code and make sure the same coding style is implemented throughout the application. --- Task 5 (Write Unit Tests) : Implement unit tests to ensure code is working as expected. --- Task 6 (Unit Testing) : Run Unit tests and ensure they pass. --- Task 7 (Documentation) : Update documentation for refactored code, buglist, and new or modified functions. --- Task 8 (System Testing) : Run system tests and ensure they pass. ---	--- Task X (Task name) -> [Who pulled task] ---	

<table>
<tr><td></td><td>Task 9 (Testing Documentation) : Update the system testing and unit testing documentation inside the testing logs.</td><td></td><td></td></tr>
<tr><td>PBI 07.0.) Invalidate IRV Ballots
<table><tr><td>PBI 07.0 - Invalidate IRV Ballots</td></tr><tr><td>As an Election Official, I want the IRV ballots to have at least half of the candidates ranked on each ballot to be considered valid, so that invalidated ballots aren't used in the election.</td></tr><tr><td>Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete):
Candidates must be ranked so that they are rounded up from .5 or above to the next higher integer value, so that the ballot is valid.
The ballots will not be invalidated at the point of collection but will need to be done when the election is run.</td></tr><tr><td>Definition of Done (what is required by the team before sending out for review):
Ballots are deemed valid or invalid
Only valid Ballots used in election
Passes unit tests
Code is refactored
Code is peer reviewed
Documentation completed
Consistent coding style throughout code
Meets all acceptance criteria conditions</td></tr><tr><td>Effort: Large</td></tr><tr><td>PBI Author(s): Ryan Mower, Hoai Bui, Emma Spindler</td></tr></table></td><td>Task 1 (Write Code) : Write code to invalidate ballots in IRV ballots when they are not of proper ballot form.

Task 2 (Peer Review Implemented Code) : Have a peer look over the implemented code that invalidates IRV ballots.

Task 3 (Refactor Code) : Go though code and refactor it, ensuring it is readable and understandable.

Task 4 (Ensure Consistent Coding Style) : Go though code and make sure the same coding style is implemented throughout the application.

Task 5 (Write Unit Tests) : Implement unit tests to ensure code is working as expected.

Task 6 (Unit Testing) : Run Unit tests and ensure they pass.

Task 7 (Documentation) : Update documentation for refactored code, buglist, and new or modified functions.

Task 8 (System Testing) : Run system tests and ensure they pass.

Task 9 (Testing Documentation) : Update the system testing and unit testing documentation inside the testing logs.</td><td></td><td></td></tr>
<tr><td>PBI 03.) Load PO Ballots</td><td>Task 1 (Write Code) : Write code to load in popularity ballots into memory.</td><td></td><td></td></tr>
</table>

PBI 03 - Load PO Ballots As an Election Official, I want the PO election ballots to be brought in a file so that the ballots can be counted and elections can be run. Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete): • Must be a .csv file • PO must be formatted as: ◦ 1st Line: PO for Popularity Only ◦ 2nd Line: Number of Candidates ◦ 3rd Line: The candidates and their party in [ ]. The name and party are separated by commas. ◦ 4th Line: Number of Ballots Definition of Done (what is required by the team before sending out for review): • Reads in ballots from CSV correctly into memory • Data is able to be worked on once in memory • Passes unit tests • Code is refactored • Code is peer reviewed • Documentation completed • Consistent coding style throughout code • Meets all acceptance criteria conditions Effort: Medium PBI Author(s)- Ryan Mower, Hoai Bui, Emma Spindler	--- Task 2 (Peer Review Implemented Code) : Have a peer look over the implemented code that loads PO ballots. --- Task 3 (Refactor Code) : Go though code and refactor it, ensuring it is readable and understandable. --- Task 4 (Ensure Consistent Coding Style) : Go though code and make sure the same coding style is implemented throughout the application. --- Task 5 (Write Unit Tests) : Implement unit tests to ensure code is working as expected. --- Task 6 (Unit Testing) : Run Unit tests and ensure they pass. --- Task 7 (Documentation) : Update documentation for refactored code, buglist, and new or modified functions. --- Task 8 (System Testing) : Run system tests and ensure they pass. --- Task 9 (Testing Documentation) : Update the system testing and unit testing documentation inside the testing logs.		

Product Backlog Items Committed to Sprint

PBI 05 - Process Multiple CSV files Into One Election
As an election official, I want the voting system to be able to take in multiple files rather than only one. This is so that we can bring in different files from different balloting locations.
Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete): ● Voting System can handle multiple CSV files to compute election ● Computes correct election results
Definition of Done (what is required by the team before sending out for review): ● Multiple CSV files are read into voting system ● System testing passes with multiple CSV files ● Code is refactored ● Code is peer reviewed ● Documentation completed ● Consistent coding style throughout code ● Meets all acceptance criteria conditions
Effort: Large
PBI Author(s): Ryan Mower, Hoai Bui, Emma Spindler

PBI 07.0 - Invalidate IRV Ballots
As an Election Official, I want the IRV ballots to have at least half of the candidates ranked on each ballot to be considered valid, so that invalidated ballots aren't used in the election.

Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete):

- Candidates must be ranked so that they are rounded up from .5 or above to the next higher integer value, so that the ballot is valid.
- The ballots will not be invalidated at the point of collection but will need to be done when the election is run.

Definition of Done (what is required by the team before sending out for review):

- Ballots are deemed valid or invalid
- Only valid Ballots used in election
- Passes unit tests
- Code is refactored
- Code is peer reviewed
- Documentation completed
- Consistent coding style throughout code
- Meets all acceptance criteria conditions

Effort: Large

PBI Author(s): Ryan Mower, Hoai Bui, Emma Spindler

PBI 03 - Load PO Ballots

As an Election Official, I want the PO election ballots to be brought in a file so that the ballots can be counted and elections can be run.

Acceptance Criteria (conditions that have to be fulfilled to ensure the story is complete):

- Must be a .csv file
- PO must be formatted as:
    - 1st Line: PO for Popularity Only
    - 2nd Line: Number of Candidates
    - 3rd Line: The candidates and their party in [ ]. The name and party are separated by commas.
    - 4th Line: Number of Ballots

Definition of Done (what is required by the team before sending out for review): ● Reads in ballots from CSV correctly into memory ● Data is able to be worked on once in memory ● Passes unit tests ● Code is refactored ● Code is peer reviewed ● Documentation completed ● Consistent coding style throughout code ● Meets all acceptance criteria conditions
Effort: Medium
PBI Author(s)- Ryan Mower, Hoai Bui, Emma Spindler