

Table of Contents (in order of priority)

PBI 1: Add candidate name list to invalid ballot file	2
PBI 2: Graphical user interface for user input	3
PBI 3: Graphical user interface to name invalidated ballot file and save	4
3-1: Give the system a file name to use to store invalid ballots	4
3-2: Use graphical interface to specify invalidated ballot file names	5
PBI 4: Create a short election report and print	6
4-1: Save short election report to a file	6
4-2: Print the short election report	7
PBI 5: Graphical system to cast ballots	8
5-1: Demo of GUI system to cast ballots	8
5-2: Remote casting of ballots	8

PBI 1: Add candidate name list to invalid ballot file

As an election official,
I want to see the candidates names listed in the invalid ballot file,
So that I don't have to look for the list of candidates elsewhere.

Acceptance Criteria:

- List of candidate names corresponding to IDs are printed in invalid ballot file

Definition of Done:

- Passes all regression tests
- Passes testing per acceptance criteria items
- Approved by product owner
- Able to show feature in demo

Effort: Small, Medium, Large, Extra Large (estimate of effort and time)

PBI 2: Graphical user interface for user input

All user input should have a graphical user interface.

As an election official,
I want to have a graphical user interface for user input,
So I know exactly where and how to select the type of election, input the number of seats, and
specify ballot file(s).

Acceptance Criteria:

- The user is presented with a GUI for inputting requirements to run an election
- Accepts plurality and STV elections
- Accepts any positive integer of seats
- Accepts any valid ballot files specified
- Works on CSELabs machines

Definition of Done:

- Passes all regression tests
- Passes testing per acceptance criteria items
- Approved by product owner
- Able to show feature in demo

Effort: Small, Medium, Large, Extra Large (estimate of effort and time)

<Hailin - Updated effort based on velocity in sprint 1>

PBI 3: Graphical user interface to name invalidated ballot file and save

The election officials should be able to name the invalidated file whatever they like and store it wherever they like. They would like to have a graphical user interface to name the file and save it

3-1: Give the system a file name to use to store invalid ballots

As an election official,
I want to be able to name the invalidated file and specify location to store it,
So that I can find the invalidated file.

Acceptance Criteria:

- The user is able to type in a file name and file path to store invalidated ballot file

Done Criteria:

- Add an option to user interface to name and store invalidated ballot file
- Function passes unit test
- Passes all regression tests
- Passes testing per acceptance criteria items
- Approved by product owner
- Able to show feature in demo

Effort: **Small**, Medium, Large, Extra Large (estimate of effort and time)

<Hailin - Update effort from medium to small, based on velocity in sprint 1. Getting user input for a file name is very doable now that we are able to set file name at run time>

3-2: Use graphical interface to specify invalidated ballot file names

As an election official,
I want to have a graphical user interface to specify invalidated ballot file name and path,
So that I can specify invalidated ballot filename and path with ease.

Acceptance Criteria:

- When specifying location and file name of invalidated ballot file, user is presented with a graphical interface (window) to input file name and select file location

Done Criteria:

- Add graphical user interface option for specifying name and path of invalidated ballot file
- Function is implemented and tested
- Passes all regression tests
- Passes testing per acceptance criteria items
- Approved by product owner
- Able to show feature in demo

Effort: **Small**, Medium, Large, Extra Large (estimate of effort and time)

<Hailin - Update effort from medium to small, based on velocity in sprint 1. The team had demonstrated the ability to implement GUI. This is just another window to get input.>

PBI 4: Create a short election report and print

4-1: Save short election report to a file

As an election official,
I want a short report with the date, type of election, candidates involved, number of seats, and winners of the election
so that the election certification officials can view the results.

Acceptance Criteria:

- A file is created after the election has been run which has the date, type of election, candidates involved, the number of seats, and the winners of the election.

Done Criteria:

- Function is implemented and tested
- Passes all regression tests
- Passes testing per acceptance criteria items
- Approved by product owner
- Able to show feature in demo

Effort: Small, Medium, Large, Extra Large (estimate of effort and time)

<Hailin - Based on experience from sprint 1. This would be fairly easy to do.>

4-2: Print the short election report

As an election official,
I want the short election report to be printed,
So that the election certification officials can view the results.

Acceptance Criteria:

- Users are presented with an option to print the election report.
- When user choose to print the election report, a report is printed on a supported device of user's choice

Done Criteria:

- Function is implemented and tested
- Passes all regression tests
- Passes testing per acceptance criteria items
- Approved by product owner
- Able to show feature in demo

Effort: Small, Medium, Large, Extra Large (estimate of effort and time)

<Bryan - unsure how to print.>

<Josh - This seems really short to me - just redirecting the results display to and ostream object instead of std::out (or we could just have results display take an ostream object and pass it std::out when we want it to print to screen). The part I'm not sure about is actually physically(?) printing the file - can we just use any system application to print a .txt file it creates or we actually have to set up working with a printer?>

<Hailin - Medium>

<Colin - Medium>

PBI 5: Graphical system to cast ballots

5-1: Demo of GUI system to cast ballots

As an election official,
I want a graphical user interface to cast a ballot,
So that votes can be sent to the voting system directly without a ballot file.

Acceptance Criteria:

- The user is presented with a GUI for casting a ballot
- Ballot is sent to voting system to be processed

Definition of Done:

- Function is implemented and tested
- Passes all regression tests
- Able to show feature in demo

Effort: Small, Medium, Large, Extra Large (estimate of effort and time)

<Colin - Large>

<Bryan - I agree that this is large. We need to update the ballot processor to have more items for error checking / handling and we need to add user displays and interfaces.>

<Hailin - Large>

<Josh - Large>

5-2: Remote casting of ballots

As an election official,
I want the Voting System to work as a backend to the Vote casting GUI,
So that ballots can be cast on remote machines and counted on a central server.

Acceptance Criteria:

- The voting system can add ballots in real time
- The voting system can be forced to stop accepting ballots
- An election can be run once the system is no longer accepting ballots

Definition of Done:

- Function is implemented and tested
- Passes all regression tests
- Able to show feature in demo

Effort: Small, Medium, Large, Extra Large (estimate of effort and time)

<Colin - Large>

<Bryan - I agree that this is large. We need updated ballot processor, error handling, user GUI and methods to tell the system when we are done collecting ballots.>

<Hailin - Large>

<Josh - Large>

CSCI 5801

P2 - Product Backlog (updated 5/1/20)

Team 3: Archer, Baker, Kluegel, Spitzer-Resnick
deak0007, bake1358, klue0037, spitz123