Initial Product Backlog
Team 26
Ge Yu yuxx0851
Hongyu Zhu zhu00091
Songyan Wu wuxx1331

1.

As a product owner I want both CPL and OPL algorithms works fully as described So that the project could be running without bugs
Acceptance Criteria: ● Unit testing codes finished ● System tes ● The program can be runned Definition of Done: ● Passes all regression test ● Passes testing per acceptance criteria items ● Approved by the team ● Return the correct results
Effort:Medium

2.

As a user I want to use a GUI So that will be more easy for me to use the the software
Acceptance Criteria: ● There are no bugs in the case ● Graphic APIs(Such as OpenGl) are applied ● Validate an executable file Definition of Done: ● Passes all regression test ● Passes testing per acceptance criteria items ● Approved by the team ● User can use the GUI to run the program
Effort: Large The GUI needs some APIs to be designed such as OpenGl or Mingfx, and it needs to be designed with interactive functions for users to search or select. Also, the searching algorithm takes time to decide, with either built-in search function or self-implemented function.

3.

As a user I want to select a file through a list instead of using the file always in the same directory So that every time when there is a new file, I don't have to put them into certain directory, only need to search or select the name after the program is running.

Acceptance Criteria:
    ● Searching Algorithms are implemented
    ● Selecting Algorithms are implemented
    ● There are no bugs in this program
 	
Definition of Done:
    ● Passes all regression test
    ● Passes testing per acceptance criteria items
    ● Approved by the team
    ● User can select or search the files they want

Effort: Large The GUI needs some APIs to be designed such as OpenGl or Mingfx, and it needs to be designed with interactive functions for users to search or select. Also, the searching algorithm takes time to decide, with either built-in search function or self-implemented function.

4.

As a software I want to handle the case with a large number of ballots up to 500,000 So that I can handle more realistic cases with large numbers, and won't let the users to wait too long for the results.

Acceptance Criteria:
    ● The data structure are large enough to read the large ballot files
    ● The algorithms are implemented fast enough to deal with the data
    ● There are no bugs in this program
 	
Definition of Done:
    ● Passes all regression test
    ● Passes testing per acceptance criteria items
    ● Approved by the team
    ● The program runs in a certain period of time

Effort: Medium First we need to test the program with over 500,000 ballots to see if the result returns in a timely manner, if so, we can keep the current design. Otherwise, we need to discuss if to use a new data structure, and implement the overall calculating method based on the old/new data structure.

5.

As a election official I want to see a short report in addition to the audit file. So that I can see the date, type of election, the candidates and their parties, the number of seats, and the winner(s) of the election.

Acceptance Criteria:
    ● Audit file is generated
    ● The information has been stored correctly
    ● The winner has been chosen corresponding to the voting and tie handler
 	
Definition of Done:
    ● Passes all regression test
    ● Passes testing per acceptance criteria items
    ● Approved by the team
    ● Able to name and store the audit file

Effort: Medium

6.

As a election official I want the ballots input on the screen directly into the program and stored to a CSV file, and be able to do this for one machine and have the voting machines send the information directly to the system So that the file can be saved as part of the audit with a nice user infterface

Acceptance Criteria:
    ● Validate ballots input
    ● Validate CSV file type
    ● Validate access of voting machines to the system
    ● Generate graphical user interface
    ● Generate success or failure messages after processing
 	
Definition of Done:
    ● Passes all regression test
    ● Passes testing per acceptance criteria items
    ● Required documentation completed
    ● Able to display the ballots input and store into CSV file
    ● Graphical user interface is accepted by the team

Effort: Medium

7.

As a election official I want to be prompted to enter the type of voting along with all of the information needed to create the ballot So that the file can be saved as part of the audit

Acceptance Criteria:
    ● At the start of an election
    ● Validate type of voting
    ● Validate the information needed to create the ballot
    ● Generate success or failure messages after processing
 	
Definition of Done:
    ● Passes all regression test
    ● Passes testing per acceptance criteria items
    ● Required documentation completed
    ● The election is closed

Effort: Large

8.

As a election official I want to be able to stop the ability to collect additional ballots and algorithms So that it can then be run after stopped

Acceptance Criteria:
    ● Validate additional ballots
    ● Validate additional algorithm
    ● Generate success or failure messages after processing
 	
Definition of Done:
    ● Passes all regression test
    ● Passes testing per acceptance criteria items
    ● Required documentation completed

Effort: Large

9.

As a election official
I want the ability to name audit file and choose where to store
So that renaming and restoring won't be necessary

Acceptance Criteria:
- Audit file is generated
- Validate if name is duplicated
- Validate if directory is valid

Definition of Done:
- Passes all regression test
- Passes testing per acceptance criteria items
- Required documentation completed
- Able to name and store the audit file

Effort: Medium