PBI Team #22

IN PROGRESS: *(priority stack is from top to bottom or the number of PBI, e.g priority of #2 > priority of #3)*

#10: Audit file includes related information

As <Election Official>
I want my audit files from the output of the voting system to include date, type of election (OPL or CPL), the candidates and their parties, the number of seats, and the winner(s) of the election. So that, the election official can get all the information that they need.

Acceptance Criteria:

- Audit files create successfully and can be open
- Test for all data needed is included in the audit files
- All data in the audit files are in an easy reading format

Effort: Small

Status: Not Started

#11: A short report is printed out and given to the election officials

As <Election Official>
I want a short report that can be printed and given to the election certification officials, so that the election officials can have the summarize report of the necessary information.

Acceptance Criteria:

- Short report prints out in an easy to read format.
- Information print out is tested
- The short report sent to the election officials successfully

Effort: Small

Status: In progess

#12: Test time manner

As <Developer>,
I want the program to be testable on runtime and efficiency to handle

more ballots and return the election result, so that when the election official runs the voting system in a timely manner with expectation upwards of 500,000 ballots.

Acceptance Criteria:

- An optional function to test the total run time of the program.
- The function can be specified as an option flag when calling ./votingsystem
- Run time < 10 second

Effort: Small

Status: In progess

#13: Input Ballot in Terminal

As <Election Official>,
I want to be able to enter ballots on the screen directly to a program and store it into a CSV file for the audit process running later. So I can create input ballots for the voting system.

Acceptance Criteria:

- Able to enter the type of voting (OPL or CPL)
- Able to enter all of the information needed to create the ballot (e.g ids, votes, ...)
- The number of ballots after the election was closed will not be counted.

Effort: Medium

Status: Not Started

#14: Handle errors for input ballots

As <Developer>,
I want erroneous inputs from the ballot file to be detected and handled, without causing a crash, so that the voting system can run successfully.

Acceptance Criteria:

- Passing tests of incorrectly formatted inputs without crashing.
- Notification of the error to the user at runtime.

Effort: Medium

Status: Not Started

#15: Able to stop input ballot from terminal

As <Election Official>,
I want to be able to input ballots live from the terminal and be able to stop inputting when needed, so that the election official can be able to stop collecting additional ballots and the algorithm would then be run.

Acceptance Criteria:

- A way to detect if the user is attempting to stop inputting and run the program.
- A way for the user to input another ballot after already having inputted one.

Effort: Medium

Status: Not Started

ACHIEVED:

#1. BUG 1

- As the product owner, I want the OPL audit process to get fixed when tie occurs. In other words, the current tie handling functionalities of OPL need to be fixed, so the election can operate properly.

- Acceptance Criteria:

- Pass ALL unit test of OPL tie handling. (Unit test for Audit() function)
- Pass ALL functional/integration test with input has a tie occurs. (Input_OPL_tie_1.cvs, Input_OPL_tie_2.cvs,...)
- Tie handling is fairly random.
- All errors in Audit() and TieHandling() are fixed if having any.

- Effort: Small

- Status: Finished

#2: BUG 2

- As the product owner, I want the CPL audit process to get fixed when tie occurs. In other words, the current tie handling functionalities of CPL need to be fixed, so the election can operate properly.

- Acceptance Criteria:
  - Pass ALL unit test of CPL tie handling. (Unit test for Audit() function)
  - Pass ALL functional/integration test with input has a tie occurs. (Input_CPL_tie_1.cvs, Input_CPL_tie_2.cvs,...)
  - Tie handling is fairly random.
  - All errors in Audit() and TieHandling() are fixed if having any.

- Effort: Small

- Status: Finished

#3: BUG 3

- As the product owner, I want the report.ExportToAuditFile() function for OPL to get fixed. Currently, the function is NOT exporting enough information (e.g how many votes a candidate or party had, order of ballot of being received,...). So, when the function gets fixed, the election report can provide correct information to users.

- Acceptance Criteria:
  - Pass ALL unit test of OPL's Report Export To File. (Unit test for ExportToFile() function)
  - Developers provide a consistent "content" format for ALL OPL output files.
  - Developers provide a consistent "name" format for ALL OPL output files.
  - All errors in ExportToFile() are fixed if there are any.
  - Include votes a candidate or party had
  - Include order of ballot of being received

- Effort: Small

- Status: Finished

#4: BUG 4

- As the product owner, I want the report.ExportToAuditFile() function for CPL to get fixed. Currently, the function is NOT exporting enough information (e.g how many votes a candidate or party had, order of ballot of being received,...). So, when the function gets fixed, the election report can provide correct information to users.

- Acceptance Criteria:
  - Pass ALL unit test of CPL's Report Export To File. (Unit test for ExportToFile() function)
  - Developers provide a consistent "content" format for ALL CPL output files.
  - Developers provide a consistent "name" format for ALL CPL output files.
  - All developers MUST look at the output files manually and agree that is the correct format.
  - All errors in Export To File are fixed if having any.
  - Include votes a candidate or party had
  - Include order of ballot of being received

- Effort: Small

- Status: Finished

#5: BUG 5

- As the product owner, I want the report.Display() function for OPL to get fixed. Currently, the function is NOT displaying enough information and has some errors (e.g column not aligned, missing information). So, when the function gets fixed, the election report can provide correct information to users.

- Acceptance Criteria:
  - Pass ALL unit test of OPL's Report Display. (Unit test for Display() function). Most of the time, those tests are manual tests.
  - Developers provide a consistent "content" format when displaying the OPL voting type.
  - All missing information are included (See description document for more information about what should be included while displaying)
  - All errors in Displaying are fixed if having any.
  - Column are aligned

- Effort: Small

- Status: Finished

#6: BUG 6

- As the product owner, I want the report.Display() function for CPL to get fixed. Currently, the function is NOT displaying enough information and has some errors (e.g column not aligned, missing information, POINTER EXCEPTION occurs). So, when the function gets fixed, the election report can provide correct information to users.

- Acceptance Criteria:
  - Pass ALL unit test of CPL's Report Display. (Unit test for Display() function). Most of the time, those tests are manual tests.
  - Developers provide a consistent "content" format when displaying the CPL voting type.
  - All missing information are included (See description document for more information about what should be included while displaying)
  - All errors in Displaying are fixed if having any.
  - Fix POINTER EXCEPTION (occur either in Audit() or Display())
  - Column are aligned

- Effort: Small

- Status: Finished

#7: BUG 7

- As the product owner, I want the UNIT TESTs and SYSTEM TESTs to be provided to cover ALL possible scenarios and functionalities (tests were missed from Project1). So, the voting system can be fully tested.

- Acceptance Criteria:
  - ALL possible test cases are provided.
  - ALL possible input for test cases are provided.
  - Testing log file provides and includes descriptions of each test case.
  - Tests for Audit() are included
  - Tests for HandlingTie() are included

- Effort: Medium

- Status: Finished

#8: Search Input File and GUI search

As <Election Official>
I want to be able to search through the file structure for a CSV file that does not have to be in the same directory,so a user can have the ability to select a file that they wanted in any directories.

Acceptance Criteria:

- Graphic Interface is implemented and easy to use
- Graphic Interface has the feature to filter the only files with the selected file extension.
- Test the return of Search to make sure the file selected is corrected

Effort: Large

Status: Finished

#9: GUI name and save

As <Election Official>
I want to be able to name the audit file and choose where to store the file. In order to do this, it has to be done using some kind of a user graphic interface, so that I can easily find the audit file later on if needed.

Acceptance Criteria:

- File created with the chosen name and chosen location successfully.
- The graphic interface is implemented and easy to use
- Filename condition tested for only allow characters and ask the user to re-enter if the file name is violated rules of the filename

Effort: Medium

Status: Finished