Sprint Backlog

Team 22 | December 3, 2019

Assignee:

- Khoa Tran
- Berni D
- Hon Kwan Sin
- Danet Chheng

Priority:

- High

User stories	To do	In progress	Complete

#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: Not Started

To do: Update UnitTest

In progress: Fix Audit() OPL when tie occur, make sure Audit() are fully tested

---

#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: Not Started

To do: Update UnitTest

In progress: Fix Audit() CPL when tie occur, make sure Audit() are fully tested

---

#3. BUG 3

- As the product owner, I want the report.ExportToFile() 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,...). 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 having any.
  - Include votes a candidate or party had
  - Include order of ballot of being received

- Effort: Small

- Status: Not Started

To do: Fix ExportToFile() OPL, include votes a candidate or party had, order of ballot of being received

To do: Check Automatic &Manual Test

#4: BUG 4

- As the product owner, I want the report.ExportToFile() 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,...). 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 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: Not Started

To do: Fix ExportToFile() OPL, include votes a candidate or party had, order of ballot of being received

To do: Check Automatic &Manual Test

#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). 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: Not Started

Complete: Fix OPL Display

Complete: Does Manual Test

Complete: Change the Display Format

#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). 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: Not Started

Complete: Fix CPL Display

Complete: Does Manual Test

Complete: Change the Display Format

#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 is provided and included descriptions of each test case.
  - Tests for Audit() are included
  - Tests for HandlingTie() are included

- Effort: Medium

- Status: In progress

Complete: Adding Unit Tests Implementation

Complete: Adding System Test Implemetation

Complete: Adding Unit Tests Inputs

Complete: Adding System Test Inputs

#8: Search Input File and GUI search

As <Election Official>
I want to be able to search for a CSV input file and use a nice graphic interface to search for the input file.

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: Not Started

To do: Impementing GUI functionalities (e.g. button.OnSelect())

To do: Manually testing GUI

To do: UnitTest and Test Logs

In progress: Design GUI (.xml or using Graphical Library) for File Search

Complete: Documentation

Complete: Error Checking if File Selected is not in Correct Format (VotingSystemController)

Design GUI *Impementing*

#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. 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: Not Started

Cards:
- Design GUI (.xml or using Graphical Library) for Naming File and Save File
- Impementing GUI functionalities (e.g. button.OnSelect())
- Documentation (Doxygen)
- Error Checking if File Saved is Existed
- Manually testing GUI
- Automatic testing and Test Logs

#10: Audit file includes related information

As <Election Official>
I want my audit files from the output of the voting system included date, type of election, the candidates and their parties, the number of seats, and the winner(s) of the election.

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

Cards:
- Adding in Report () date, type of election, candidates, parties, number of seats, and winners
- Update UnitTest for Audit()
- Update UnitTest for Report()
- Update the Audit() that send additional information to Report

#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.

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: Not Started

Cards:
- Adding UnitTesting and Test Log
- Manully Test the Short Report
- Implementing ExportShortReport() for OPL
- Implementing ExportShortReport() for CPL
- Documenting (Doxygen)

#12: Test time manner

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

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

Effort: Small

Status: Not Started

Cards:
- Adding Automated Unit Test and Test Log
- Adding Time Block Checking function for Audit()

#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 CVS file for 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 ballots information on days after the election was closed will not be counted.

Effort: Medium

Status: Not Started

Cards:
- Implementing function that users can type OPL and CPL in Terminal
- Testing the function to be able to input CPL and OPL
- Testing ballots inputs in Terminal
- Implementing the function that handle after election closed, so ballots will not counted
- Saving input to CSV
- Unit Test and Test Log
- Implementing function that user can input the ballots in Terminal

#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.

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

Effort: Medium

Status: Not Started

Cards:
- Testing of incorrectly formated inputs without crashing
- Implementing the function that checking errors for input ballots in Terminal
- Testing runtime errors
- Handling runtime input errors. When errors occur, notify user.

#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 need be.

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

Effort: Medium

Status: Not Started

Cards:
- Implementing function that can stop/break users from input ballots in Terminal
- Testing Detect ballots input