Software Requirements Specification

for

Election Calculator Software

Version 1.8 approved

Prepared by Christopher Atherton, Zaid Nolley, and Zach Ross

Team19

2/12/2024

Table of Contents-

Revision History

Name	Date	Reason For Changes	Version
Christopher	02/04/24	Did section 1 and 2	1.1
Zaid	02/07/24	Completed 1st drafts for sections 3 and 5	1.2
Zachary	02/07/24	Did most of 1st draft for section 4	1.3
Christopher, Zaid, Zachary	2/07/24	Various revisions	1.4
Zachary	02/11/24	Completed section 4 and Glossary A	1.5
Zaid	2/12/24	Edited sections 3 and 5 and formatted them for submission.	1.6
Christopher, Zaid, Zachary	2/12/24	Various revisions, use cases	1.7
Christopher	2/12/24	Final touches	1.8
Zachary	3/1/24	Fixed missing requirement	1.9

1. Introduction

1.1 Purpose

This document provides details of the requirements and specifications of the Election Calculator Software (ECS). It is a standalone, new release, version 1.0. The scope is a standalone system for calculating the results of OPL (Open Party Listing) and CPL (Closed Party Listing) elections.

1.2 Document Conventions

Not applicable

1.3 Intended Audience and Reading Suggestions

This document is intended for any user of the software such as an election official or auditor. As well as anyone who may be interested in an election and the design of the software that calculates the winners.

If you are a user of the software you should primarily read through sections 2 and 4 which detail the overall description and features of the software. You should especially read subsections 2.6 and 4.1. Otherwise, if you are looking into the design of the software, read sections 3 and 5 which detail the different requirements the software performs under.

1.4 Product Scope

This product is an automatic ballot counter and seat allocator for a CPL or OPL election. The software takes manual input of a .csv file formatted correctly to list the parties, candidates, and every ballot of an election. The software allows the process of counting votes and allocating seats to candidates to be automated.

1.5 References

Not applicable

2. Overall Description

2.1 Product Perspective

This product is designed as a component of a CPL or OPL election which is looking to automate the vote counting and seat allocation process. As a component of these elections, this software requires security to ensure no tampering is done and votes are correctly counted. The software relies on the .csv file from the election to be used.

2.2 Product Functions

- Works for 2 election types - This software can only be used to automatically process a CPL or OPL election.
- Count Ballots - This software will count the number of votes each party or candidate receives as listed in the given .csv file.
- Allocate Seats - This software will allocate the seats available to the candidates specified in the .csv based on the election's respective seat allocation algorithm.
- Displays Election Results - This software will display the results of the election on the user's screen, as well as more detailed results in an output file.
- Command Line Ballot Input - This software will allow ballot input through the Command line.
- Terminal GUI (Graphical User Interface) Ballot input - This software will allow ballot input through an intractable graphical user interface.

2.3 User Classes and Characteristics

The most important users of this product are people calculating results of elections, be they election officials or auditors.

Another class of users is product testers who are likely to use the software at a much higher rate. They will also prefer the Command line input for easier testing.

2.4 Operating Environment

This software will operate on any UNIX operating system. It will be run through the terminal with input from a .csv file properly formatted for either a CPL or OPL election. The software will output results to the screen and to a file.

2.5 Design and Implementation Constraints

- Software must process up to 100,000 ballots in less than 4 minutes.
- A single program must handle all use cases.
- File format must adhere to the specified OPL or CPL format.
- The software can only process two election types, OPL and CPL.
- Input files will be read-only to ensure election security.
- The software will be coded in C++.
- The software must provide accurate election results.
- In the event of a tie, a candidate will be fairly chosen through random chance.

2.6 User Documentation

A README file describing how to use the software will be provided.

2.7 Assumptions and Dependencies

It is assumed that any files put into the software will be formatted correctly with valid data of the ballots.

It is assumed any users have a basic understanding of file management and UNIX terminals.

3. External Interface Requirements

3.1 User Interfaces

Main Path

1. The typical user will launch the software by clicking on an executable file that is stored in the file manager.

2. The program will launch on a terminal window and has only text display and input.

3. The program prompts the user to enter the .csv file name that contains the ballot results and will display a generic example input (e.g. “/election11-2024.csv/”). At startup, the program also displays a keyboard option to quit the program (e.g. “To exit type “q” and press enter”.)

4. In the event of an invalid input (e.g. path mistake, typo, wrong file type), the terminal will output a message indicating that an error occurred and what type of error (otherwise a generic/unknown error message will be displayed). Afterwards, the terminal will then prompt the user to enter the .csv ballot file or exit the program.

5. Once the program executes a ballot file, a message will display to the terminal indicating that the results are being calculated.

6. The election results will be displayed simultaneously on the terminal followed by a message indicating that an audit file has been created, its name, and the location of the audit file in the file manager. The terminal will ultimately display that the program has been terminated.

Alternative Path

1. A user may use the terminal and maneuver to the location of the software executable.

2. The user then types in a “./” and the executable file name.

3. If the user presses enter, the program launches and is identical to step 2 in the main path.

4. Otherwise the user can add a space and then type in the name of the .csv ballot file and then press enter, at which point the program runs identically to the main path at step 4.

3.2 Hardware Interfaces

The program is designed and guaranteed to be runnable on the University of Minnesota Twin Cities CSE lab machines (circa spring 2024) which have the following specifications:

- CPU: Intel® Core™ i7 @ 2.5 GHz (x8)
- 32 GB RAM (Random Access Memory)

These machines can be locally accessed or accessed remotely through SSH to run the program.

3.3 Software Interfaces

The software is designed to be run on Linux Ubuntu 20.04. The computer running the software must have terminal access, a file manager to take input files/store audit files, be able to run C++ executable files and be able to display text files.

3.4 Communications Interfaces

Not Applicable.

4. System Features

This section details the Election Calculator's primary function. The goal is to explain its usage.

4.1 Ballot Input Via GUI

4.1.1 Description and Priority

The GUI will allow election officials to enter the ballot by inputting its filename. The system will then automatically extract all information from the ballot.

Priority High

4.1.2 Stimulus/Response Sequences

a. The election official executes the program on a computer meeting the required specifications
b. The program prompts the election official, "Enter the filename of the ballot."
c. All of the information on the ballot is automatically extracted by the program

4.1.3 Functional Requirements

REQ-1: Filename input via GUI/text.
REQ-2: Extract all information for the election from the ballot.
REQ-9: Handle invalid file names.

a. If a user inputs a filename that is not in the directory the program will prompt the election official, "Unable to find *example_filename* please check that the file is in the same directory as the Election Calculator and that you correctly spelled the filename."

4.2 Ballot Input via Command Line

4.2.1 Description and Priority

The command line will allow testers to enter the filename of the ballot to be calculated. The system will then automatically extract all information from the ballot.

Priority High

4.2.2 Stimulus/Response Sequences

a. The tester executes the program and inputs the filename in one line on the command line.

   example (*$ example_program.exe example_file_name.cvs)*

b. The system will then automatically extract all information from the ballot.

4.2.3 Functional Requirements

REQ-3: Filename input into the command line using bash
REQ-2: Extract all information for the election from the ballot.
REQ-9: Handle invalid file names.
   a. If a user inputs a filename that is not in the directory the program will prompt the tester, "Unable to find *example_filename* please check that the file is in the same directory as the Election Calculator and that you correctly spelled the filename."

4.3 Tie Breaking via Fair Coin Flip

4.3.1 Description and Priority

This function allows ties in the election calculations to be automatically decided via a fair and unbiased coin flip. The program automatically selects which form of tie-breaking it is performing, OPL or CPL, based on the information collected from the ballot.

Priority High

4.3.2 Stimulus/Response Sequences

a. A tie is encountered during the election calculations.
b. The program automatically runs a coin-flipping simulation to determine the outcome of the tie event according to the information provided on the ballot. If OPL then it follows the conventions of OPL and if CLP then CPL conventions.

4.3.3 Functional Requirements

REQ-4: When a tie occurs, determine the seat assignment according to OPL conventions, based on the coin flip outcome.
REQ-5: When a tie occurs, determine the seat assignment according to CPL conventions, based on the coin flip outcome.
REQ-10: A fair coin flip can be done with a random number generator.

4.4 Calculating Election Results

4.4.1 Description and Priority

The system will extract the data from the file and store the voting totals in an internal data structure. The system will then use that information to assign seats based on which voting system is specified in the file, OPL or CPL.

Priority High

4.4.2 Stimulus/Response Sequences

a. The program has been executed and the filename input.
b. The program then extracts the data and counts the votes by reading in the lines based on the agreed-upon file format.
c. The program automatically assigns seats based on the voting system specified in the file, OPL or CLP.

4.4.3 Functional Requirements

REQ-2: Extract all information for the election from the ballot.
REQ-6: Mechanism to process ballots through the file.
REQ-11: One program to handle all election types.

4.5 Post Election Results

4.5.1 Description and Priority

The system displays the election results. It also creates an audit file that contains all the information displayed on the election results.

Priority Low

4.5.2 Stimulus/Response Sequences

a. The system finishes calculating the winners of seats.
b. The system displays all election information to the display according to the agreed-upon format.
c. The system creates an audit file and populates it with all election information, according to the agreed-upon format.

4.5.3 Functional Requirements

REQ-7: Create Audit File
REQ-8: Display all Information on the screen

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Designed to run on Linux Ubuntu 20.04, the University of Minnesota Twin Cities CSE lab machines with

- CPU: Intel® Core™ i7 @ 2.5 GHz (x8) CPU
- 32 GB RAM

These machines should be able to process and produce the results from a .csv ballot file containing 100,000 ballots in under 4 minutes. Machines with lower specifications may take longer to calculate and display results.

5.2 Safety Requirements

1. Valid ballots and a correctly formatted input file are preconditions for this software and legitimate and correct results depend on this; this is ensured by election officials before the use of this software.
2. The passed-in .csv ballot file will not be able to be edited or written to by this software, only read.
3. The audit file produced after an election is calculated should have a name that indicates what file was initially passed in and at what time the election was calculated.
4. Elections must only be run once to maintain their fairness (especially in the case of a tie) and thus the program terminates after a single election is run.

5.3 Security Requirements

The software itself does not have any specific security requirements to run the application.

5.4 Software Quality Attributes

Due to a lack of a custom GUI or icon shortcut, the user of this program should be familiar with the basics of a typical computer file management system and be able to maneuver through one to launch and use the program. Additionally, the user should understand what a file path is and how to read and type one so that the ballot file can be passed in and the audit file can be accessed.

Users are also expected to have an understanding of how Open Party Listing and Closed Party Listing elections work to interpret the displayed results and the contents of the audit file.

5.5 Business Rules

The software is not inherently restricted in usage to anyone and it may be used by whomever has access to it and has the properly formatted .csv ballot files.

6. Other Requirements

6.1 Use Cases

Name	Extract all information from the ballot file.
ID	UC_001
Description	When inputting a ballot file, users can extract the information including election type and candidate info, as well as the ballots themselves.
Actors	System
Organizational Benefits	Allows ballot files to be processed automatically.
Frequency of Use	Any time a ballot file is input.
Triggers	User inputs ballot file.
Preconditions	Software is running, a ballot file has been input.
Postconditions	The file's information has been extracted and winners can then be calculated.

Main Course	1. User runs the software. 2. User inputs file. (EX1) 3. Information from the file is extracted.
Alternate Courses	N/A
Exceptions	EX1 User inputs file. 1. File cannot be found. 2. System asks for the file to be input again. 3. If a file cannot be found return to EX1 4. If a file is found, move on to main course 3.

Name	Pass ballot file name into the program
ID	UC_002
Description	The user wants to run the election calculator software directly from the terminal by including the .csv ballot file name with the typed-in program executable file.
Actors	Testers
Organizational Benefits	Allows easier and more streamlined testing for designers of the program.
Frequency of Use	Significant usage during the design and testing phases of the software, followed by little usage by most users of the ultimate product.
Triggers	The user/tester runs the software
Preconditions	The user is in the directory of the executable software file and the .csv ballot file exists in the same directory.
Postconditions	The program is run.
Main Course	1. The user in the terminal maneuvers to the directory of the election calculator software executable file. 2. User types "./" followed by the executable file name, followed by a space, and then types the .csv ballot file name and then presses enter (or return on some keyboards). 3. The program is run.

Exceptions	System error occurs 1. User is notified that an error occurred and of what type.

Name	Pass ballot file name into the program via GUI.
ID	UC_003
Description	Users can manually input the name of a file to be processed.
Actors	Election Officials/ most users.
Organizational Benefits	Allows pre-existing ballot files to be processed.
Frequency of Use	Whenever the software is used.
Triggers	When software is run.
Preconditions	Software is run and asks for file input.
Postconditions	Software can now read the ballot file.
Main Course	1. Software is ran through a UNIX terminal with the "./" command 2. Software prompts users to input the file name. (AC1) 3. File is input by the user.
Alternate Courses	AC1 Software prompts users to input the file name. 1. User uses an alternate file directory system to maneuver to the directory containing the software executable 2. The user launches the program by double-clicking or selecting the executable file and pressing enter. Return the user to main course step 2
Exceptions	N/A

Name	Handle tie in CPL election
ID	UC_004
Description	A tie in a CPL election results in a random, fair decision between two or more parties.

Actors	System
Organizational Benefits	Allows for ties to be broken randomly and fairly.
Frequency of Use	Whenever a tie in a CPL election occurs, which will rarely occur in real use cases.
Triggers	When allocating seats a tie occurs between parties.
Preconditions	A tie between two or more parties occurs.
Postconditions	The tie is broken through fair random chance and the seats are allocated.
Main Course	1. A tie occurs. 2. The tie is broken by choosing one of the tied parties randomly until no tie is left.
Alternate Courses	N/A
Exceptions	N/A

Name	Handle tie in OPL election
ID	UC_005
Description	A tie in an OPL election results in a random, fair decision between two or more candidates.
Actors	System
Organizational Benefits	Allows for ties to be broken randomly and fairly.
Frequency of Use	Whenever a tie in an OPL election occurs, which will rarely occur in real use cases.
Triggers	When allocating seats a tie occurs between candidates.
Preconditions	A tie between two or more candidates occurs.
Postconditions	The tie is broken through fair random chance and the seats are allocated.

Main Course	1. A tie occurs. 2. The tie is broken by choosing one of the tied candidates randomly until no tie is left.
Alternate Courses	N/A
Exceptions	N/A

Name	Process header information in the ballot file
ID	UC_006
Description	The header of a ballot file which contains information on the election type, parties, and candidates is processed.
Actors	System
Organizational Benefits	Allows ballots and information about the election to be processed separately.
Frequency of Use	Whenever a file is input to the software.
Triggers	Whenever a file is input to the software.
Preconditions	A file of either election type is input.
Postconditions	The header is processed and the software records the election type and party information.
Main Course	1. User inputs file into the software. 2. Header is processed. 3. Election type and party information are recorded.
Alternate Courses	N/A

Exceptions	N/A

Name	Process Ballots
ID	UC_007
Description	Ballots are processed and tallied to record the votes received for parties or candidates.
Actors	System
Organizational Benefits	Allows vote counting to be automated.
Frequency of Use	Whenever a ballot file is processed.
Triggers	A ballot file is input into the software.
Preconditions	Header has been processed, the file is open, and the election type is known.
Postconditions	Votes for parties and candidates are counted and recorded.
Main Course	1. Software counts each line of the ballot file as a singular vote, counting and recording the vote for the respective party or candidate. 2. Software reaches the end of the file and is finished counting votes.
Alternate Courses	N/A
Exceptions	N/A

Name	Create Audit File
ID	UC_008
Description	After winners are decided and seats are allocated, the software creates an audit file containing information on the results of the election.

Actors	System
Organizational Benefits	Allows results to be on a recorded file.
Frequency of Use	Whenever an election is processed.
Triggers	Seats have finished being allocated.
Preconditions	Seats have finished being allocated.
Postconditions	An audit file is created containing the results of the election.
Main Course	1. Election has been processed. 2. Software creates an audit file and opens it for writing. 3. Software writes the election type, winners, and allocation of seats to the audit file. 4. Software closes the file and saves it to the program directory.
Alternate Courses	N/A
Exceptions	N/A

Name	Display Results
ID	UC_009
Description	After winners are decided and seats are allocated, the software displays the results of the election on the screen.
Actors	System
Organizational Benefits	Allows results to be immediately displayed on screen.
Frequency of Use	Whenever an election is processed.
Triggers	Seats have finished being allocated.

Preconditions	Seats have finished being allocated.
Postconditions	The results of the election are displayed on the screen.
Main Course	1. Election has been processed. 2. Results are displayed on the screen.
Alternate Courses	N/A
Exceptions	N/A

Name	One Program All Elections
ID	UC_010
Description	One program to handle all election types.
Actors	System
Organizational Benefits	This use case will allow users to only need to input the name of the file. The system will automatically decide the election based on information in the ballot. It will automatically select the appropriate type of election.
Frequency of Use	Frequent
Triggers	The program is being executed via a command-line with an accurate file name. Or the program being executed, prompting the election official via GUI, and the file name being correctly entered.
Preconditions	Valid filename being provided, open read-only file.
Postconditions	The system automatically selects which election type to run. The results are displayed on the screen and an audit file has been generated.
Main Course	1. The user executes the program and input a file name. 2. The system automatically extracts all information from the ballot. 3. The system uses the information to determine

Alternate Courses	N/A
Exceptions	N/A

Name	Assign Seats to winners OPL
ID	UC_011
Description	After winners are decided the system assigns seats to winners
Actors	System
Organizational Benefits	Automatic Seat Assignment of winners
Frequency of Use	Whenever an election is processed.
Triggers	Votes have finished being allocated.
Preconditions	Votes have finished being allocated.
Postconditions	The results of the election are displayed on the screen.
Main Course	3. Election has been processed. 4. Results are displayed on the screen.
Alternate Courses	N/A
Exceptions	N/A

Name	Assign Seats to winners CPL
ID	UC_012
Description	After winners are decided the system assigns seats to winners
Actors	System
Organizational Benefits	Automatic Seat Assignment of winners
Frequency of Use	Whenever an election is processed.
Triggers	Votes have finished being allocated.
Preconditions	Votes have finished being allocated.
Postconditions	The results of the election are displayed on the screen.
Main Course	5. Election has been processed. 6. Results are displayed on the screen.
Alternate Courses	N/A
Exceptions	N/A

Appendix A: Glossary

- .csv - is a text file format
- Audit file - a file to track election results to be analyzed at a later date.
- Ballot - a form recording a voters candidate choices
- C++ - is a computer programming language
- Candidate - someone running for an elected office
- Command Line - is a method of communicating with a program
- CPL - or Closed Party Listing is a type of election system.
- CPU - or Computer Processing Unit is a hardware component of most computer systems.
- Data Structure - is a method to organize data in a program.
- Directory - is a part of a computer's file management system, usually containing other directories and/or files.

- Election - a method for selecting a candidate via voting.
- Executable Files - a file that can be run once prompted.
- Extract - to remove contents.
- GUI - or Graphical User Interface is a reason to interact with a program and for the program to display information.
- Linux Ubuntu 20.04 - is a type of operating system.
- Operating System - the software that handles most of the computer's functions, namely programs.
- OPL - or Open Party Listing is a type of election system.
- Path - relates to directory navigation. Directories link to one another this forms their “path”.
- Product Testers - often computer programmers who ensure a product in development works according to the agreed upon customer specifications.
- RAM - or Random Access Memory is a hardware component of most computer systems.
- README file - is a file describing useful or relevant information about a particular program, usually found in the program’s directory.
- Seat - a position that is filled with elected officials.
- Software - is a computer program.
- Terminal - is a program that utilizes command line prompts and inputs to allow users efficient access programs.
- UNIX - is a type of operating system.
- Use Case - is a specific scenario in which to apply a solution.

Appendix B: Analysis Models

Not Applicable

Appendix C: To Be Determined List

Not Applicable