Software Requirements Specification

for

Project Aegis

Version 1.0 approved

Prepared by <author>

<organization>

2/19/2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the runoff voting system. It will explain the purpose and features of the system, what the system will do, the constraints under which it must operate. This software will take in a csv file and determine the outcome based on the two types of voting systems: instant runoff and open party list. This document is intended for the developers of the system, the users of the system, and will be proposed to the government for approval for an upcoming election.

1.2 Document Conventions

This document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

- Programmers who are further developing and/or fixing bugs. It's recommended that they start on System Features and work their way down through the requirements.
- Election officials and government bodies who are looking for a software to manage and handle the outcome of an election

1.4 Product Scope

Aegis is a system for running elections in either Instant Runoff voting (IR) or Open Party List voting (OPL). The system to be used, IR or OPL, is determined by a file header, but the user can choose the layout of result information. This software is designed for use in normal or special elections throughout the year.

1.5 References

Project 1 –Waterfall Methodology Software Requirements Specification (SRS) Document for Voting System

https://canvas.umn.edu/courses/217913/files/18790599?wrap=1

2. Overall Description

2.1 Product Perspective

Aegis is a software being developed for use by election officials to count and compute results for normals and special elections through-out the year. It handles CSV files containing voter ballot information.

This is a class project to explore a specific pipeline in software engineering involving teamwork and product specifications. It will be developed to run on Windows, Mac OS X and Linux.

2.2 Product Functions

- Enter election data: Data files entered as CSV
- Process election data: Program processes the CSV based on OPL or IR algorithm
- Output a winner: Program prints out who the winner of the election is
- Output an audit file: Program prints out the exact steps it took to determine the winner
- Break ties with a fair RNG: Program randomly determines outcome of a tie
- Output file with data for testers: Extra data is outputted in addition to the audit file to aid testers
- Flags for testing
    - --forceTie: allows user to determine the outcome of a tie
    - --time: displays the amount of time taken from start to end of the program
-

Indicate the types of ballots

2.3 User Classes and Characteristics

Election Officials: Officials appointed by government bodies to preside over electoral precincts and ensure that elections run smoothly and without compromise. They need to input the ballots they have collected into the program in order to receive official results.

Programmers: The developers of this program will have additional commands in addition to the commands already provided with the program in order to help debug and validate the results of the program.

Testers: Testers will have the same capabilities as the election officials in addition to extra arguments that they can use in order to test the program to see the variability algorithm for speed as well as accuracy in the results.

2.4 Operating Environment

- Recommended
    - Windows
        - OS: Windows7/Windows8/Windows10
        - Processor: Xeon @ 3GHz (x12), Intel® Core™ i7 @ 3.2GHz (x6)
        - Memory: 32/64 GB RAM
        - Storage: 10 MB
    - Mac
        - OS: Mac OS X
        - Processor: Intel Core i5 3.2 GHz, Intel Xeon e5-2460 2.4ghz 20 cores, Intel Core i7-4770 3.40GHz
        - Memory: 16 GB RAM
        - Storage: 10 MB
    - Linux
        - OS: Ubuntu 20.04/18.04/16.04
        - Processor: Intel® Core™ i7 @ 1.6GHz (x3), Intel® Core™ i7 @ 900Hz (x3)
        - Memory: 32 GB RAM
        - Storage: 10 MB
- Minimum
    - Windows
        - OS: Windows7/Windows8/Windows10
        - Processor: Intel® Core™ i5 @ 800Hz (x3)
        - Memory: 32 GB RAM
        - Storage: 10 MB

- Mac
  - OS: Mac OS X
  - Processor: Intel Core i5 3.2 GHz, Intel Xeon e5-2460 2.4ghz 20 cores, Intel Core i7-4770 3.40GHz
  - Memory: 16 GB RAM
  - Storage: 10 MB
- Linux
  - OS: Ubuntu 18.04/16.04
  - Processor: Intel Xeon e5-2460 2.4ghz 20 cores, Intel® Core™ i5 @ 800Hz (x3)
  - Memory: 32 GB RAM
  - Storage: 10 MB

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

100,000 ballots in 8 minutes. See section 5.1 for more information.

This program is developed in C++.

2.6 User Documentation

Documentation will be generated in the form of html and latex files using Doxygen with doxyfiles. Github will be user version control.

See section 3.1 for instructions on how to run this program.

2.7 Assumptions and Dependencies

- We assume that the user is going to be using a CSE lab machine and that they have a UMN account.

- We assume that there are no numbering mistakes in the file and that there are no mistakes made on the voter end.
- We assume that each ballot in the file has at least one candidate and that there are no errors in the ballots.
- We assume that the user knows how to properly operate and access a CSE lab machine. They should also know basic computer skills.

3. External Interface Requirements

3.1 User Interfaces

Election Officials

In order to run this program, open up the terminal and type "./Aegis file.csv". After the winner has been determined, the program will prompt the user if they would like to produce an audit file. The user will type "Y" for yes or "N" for no. If "Y" was inputted, the output file will be produced in a pdf file in the same directory as the program and titled as the date and time it was created.

Developer/Tester

In order to run this program, open up the terminal and type "./Aegis file.csv". Extra flags can be appended such as "--forceTie". For all flags, see section 2.2. After the winner has been determined, the program will prompt the user if they would like to produce an audit file. The user will type "Y" for yes or "N" for no. If "Y" was inputted, the output file will be produced in a pdf file in the same directory as the program and titled as the date and time it was created. Any extra information will be printed out as determined by the flags.

3.2 Hardware Interfaces

The voting system does not rely on hardware components.

See section 2.4 for minimum software/hardware components.

3.3 Software Interfaces

See section 2.4 for software/hardware components.

Data will be attained from a provided CSV file but we will keep in mind that later on use of databases may be preferable.

3.4 Communications Interfaces

This program is required to be run on CSE labs through a physical connection or through a virtual connection such as Vole or SSH.

4. System Features

4.1 Run Open Party Listing (OPL)

4.1.1 Description and Priority

High Priority

The purpose of this feature is to collect the voting results from a valid csv file and then process the file to receive the results of an Open Party Listing type election. The results of the program will be displayed on the screen. An audit file will be produced when running the program in order to confirm the validity of the election results which is stored in the same directory as the program. A media file will also be produced in the same directory which contains information which can be used for media coverage.

4.1.2 Stimulus/Response Sequences

The user, in addition to running the program, will pass in a csv file from the command line when running the program. The system then processes the file with an OPL algorithm in order to receive the results of the file. An audit file and a media file will be produced in the same directory as the program. The results of the election will be displayed on the screen.

4.1.3 Functional Requirements

REQ-1(VS_003): The system takes in a csv file. If the file is invalid, the user will be prompted to run the program again with a valid csv file.

REQ-2: The system will process the file and determine which voting system to run.

REQ-3(VS_006): The system will group independent parties as a single party if there are any independent parties. See section 4.7

REQ-4(VS_004) : The system now runs the calculations using the OPL algorithm.

REQ-5(VS_001): If there is a tie, the system will break the tie by giving each candidate a random number to make sure the probability is balanced between the candidates and randomly select one. See section 4.6 for further details.

REQ-6(VS_007): The system will display the results of the election. See section 4.5 for further details.

REQ-7 (VS_002, VS_011): The system will produce an audit file and a media file in the same directory as the program. See section 4.3 and 4.4 for further details.

4.2 Run Instant Runoff (IR)

4.2.1 Description and Priority

High Priority

The purpose of this feature is to collect the voting results from a valid csv file and then process the file to receive the results of an Instant Runoff type election. The results of the program will be displayed on the screen. An audit file will be produced when running the program in order to confirm the validity of the election results which is stored in the same directory as the program. A media file will also be produced in the same directory which contains information which can be used for media coverage.

4.2.2 Stimulus/Response Sequences

The user inputs a file in the form of the csv into the command line. This triggers a response of running the results under the IR voting system. An audit file and a media file will be generated in the same directory as the program run. Results of the election will be displayed on screen.

4.2.3 Functional Requirements

REQ-1(VS_003): The system takes in a csv file. If the file is invalid, the user will be prompted to run the program again with a valid csv file.

REQ-2: The system will process the file and determine which voting system to run.

REQ-3(VS_005): The system now runs the calculations using the IR algorithm.

REQ-4(VS_001): If there is a tie, the system will break the tie by giving each candidate a random number to make sure the probability is balanced between the candidates and randomly select one. See section 4.6 for further details.

REQ-5(VS_007): The system will display the results of the election. See section 4.5 for further details.

REQ-6(VS_002, VS_011): The system will produce an audit file and a media file in the same directory as the program. See section 4.3 and 4.4 for further details.

4.3 Produce Audit File

4.2.1 Description and Priority

High Priority

The system will provide a user with an audit file at the end of running and processing the file with election information. What is provided in this audit file will be the type of voting that occurred, number of candidates, candidates, number of ballots, calculations, vote distribution for candidates, along with the winner(s) of an election and how the election progressed to that result.

4.2.2 Stimulus/Response Sequences

This feature is stimulated in response to a successful and complete run of a csv file containing the election information. In response to such stimulus an audit file will be produced with strict ordering depending on whether OPL or IR was run.

4.2.3 Functional Requirements

REQ-1(VS_003): The system must have been provided with a csv file.

REQ-2: The system must be able to successfully run with provided csv file.

REQ-3(VS_002): The system will produce an audit file with strict formatting to represent the type of voting that occurred, number of candidates, candidates, number of ballots, calculations, vote distribution for candidates, along with the winner(s) of an election and how the election progressed.

4.4 Produce Media File

4.2.1 Description and Priority

Medium Priority

Given that an election's results have been computed and finalized, it would probably be a good idea to provide this information to the public through sharing these results with media personnel. With this in mind, having a file to provide for the media (the Media File) with election results will be a feature.

4.2.2 Stimulus/Response Sequences

Given that a successful and complete computation of a csv file containing the election information is run, production of the media file will be done.

4.2.3 Functional Requirements

REQ-1(VS_003): The system must have been provided with a csv file.

REQ-2: The system must be able to successfully run with provided csv file.

REQ-3(VS_011): A resulting media file, with election results will be generated.

4.5 Display Results

4.2.1 Description and Priority

High Priority

After processing the csv file accordingly, the system will display the results of the election onto the screen. The party and name of the winner(s) will be displayed as well as the election type and number of seats.

4.2.2 Stimulus/Response Sequences

A valid csv file has been inputted and processed successfully. The results of the election have been outputted to the display.

4.2.3 Functional Requirements

REQ-1: The program has been completed successfully.

REQ-2(VS_007): The result of the election is displayed onto the screen.

4.6 Resolve Ties

4.2.1 Description and Priority

High Priority

The system will resolve any ties that come about in a fair and unbiased manner in order to facilitate a secure election.

4.2.2 Stimulus/Response Sequences

This event is stimulated when there is a tie between two or more candidates. After this, one candidate has been decided the winner.

4.2.3 Functional Requirements

REQ-1: The program has successfully run until a tie has occurred between any number of candidates.

REQ-2: The system will assign consecutive integers starting from 0 to any candidate that is part of the tie.

REQ-3(VS-001): The system will use a random number generator (RNG) that includes all integers assigned to candidates and excludes any integers not used.

REQ-4: The result of this RNG will be the candidate who wins the tie, and the system will treat them as such.

4.7 Group Independents

4.2.1 Description and Priority

High Priority

The purpose of this feature is to simplify the algorithmic process when running an Open Party Listing based election. The system after receiving the document, if it determines that it is an OPL based election (See section 4.1), before running the calculations will group all the independent parties into a single party.

4.2.2 Stimulus/Response Sequences

This feature will be activated when two conditions are met. The election type is Open Party Listing and there is at least one independent party. The response is that if the conditions are met, the system will group the independent parties into one party.

4.2.3 Functional Requirements

REQ-1(VS_003): The system determines after reading the csv file that the election type is Open Party Listing.

REQ-2: The system determines after reading the csv file that there is at least one independent party.

REQ-3(VS_006): The system groups all of the independent parties into one party for the use of OPL. See section 4.1

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Needs to be able to process 100,000 ballots within eight minutes. This is to make sure that the program runs quickly enough to produce the results of the election in an orderly time.

5.2 Safety Requirements

No safety requirements.

5.3 Security Requirements

No security requirements as the document required for this system.

5.4 Software Quality Attributes

Easy to use, so the election officials are able to easily use the program.

Portability, so many different computers can run the program in their respective regions for the elections.

Accurate, the results of the program are guaranteed to be correct as it is important for an election to have absolute reliability in its results.

5.5 Business Rules

Only election officials can use the program outside of the developers and testers. Election officials will use the program in order to process the election ballots. Election officials may share the results to the media afterwards.

6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: To Be Determined List