Software Requirements Specification

for

Voting Count System (VCS)

Version 1.0 approved

**Prepared by Tobias Moszer, Tanner Skluzacek,
Varun Reddy, Jonathan Leibovich**

University of Minnesota

February 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 Voting Count System (VCS). More specifically, this system will be capable of calculating the results of both Instant run-off and Open Party-List Elections. It will explain the purpose and features of the system, the interfaces of the system, and the constraints under which it will operate. This document is for election officials, programmers, and testers to be referenced when using this system.

1.2 Document Conventions

The IEEE template was used for the system requirements specification document.

1.3 Intended Audience and Reading Suggestions

The intended audience of readers for this document are programmers, testers, and users (election officials). This document contains the implementation strategies and methods used in creating a secure and reliable voting system. Programmers will want to focus more on the system features section, once they have a clear understanding of the overall system. Testers will want to focus more on the Use Cases. They need to make sure the system does what it is supposed to do, so they need a clear understanding of its intended behaviors. Users will need a basic understanding of the system, with emphasis on external requirements like safety, security, software, and hardware, making sure they are correctly operating the system to ensure reliability.

1.4 Product Scope

The VCS is a product that is able to calculate results of an election, given the voting data. It is able to do this for both Instant Runoff Voting and Open Party List Voting. The goal of this product is to accurately and quickly calculate the results of a given election for two types of voting formulas. The intended use is for election officials to quickly and accurately conclude the outcome of an election. Files will also be created to share the results with the media and to audit the validity of the election outcome.

1.5 References

<1> - IEEE Template for System Requirement Specification Documents: https://goo.gl/nsUFwy

<2> - FairVote Explanation for Instant Runoff Voting:
https://www.fairvote.org/rcv#how_rcv_works

<3> - FairVote Explanation for Party List Voting:
https://www.fairvote.org/how_proportional_representation_elections_work

2. Overall Description

2.1 Product Perspective

The VCS is a system that reads a file containing election data, calculates the election results, and displays the final results of the election for the election official that is using the system. The product only relies on a given CSV file that is formatted properly to read and produce the results. This project is developed by a team of four to be run on Mac OS, Ubuntu, and Windows operating systems. The project is independent of any other larger system, therefore all of the voting must be done beforehand and formatted into a CSV file. This product was developed mainly for election officials to quickly tally the votes of either an Open Party List or Instant Runoff Voting election.

2.2 Product Functions

- Bring File into System (Use case 4.1)
    - Prompt the user for the name of the election file using the OS's terminal. Next, attempt to open the file within the software. If unsuccessful, prompt the user again.
- Preprocessing Data from Ballot File (Use case 4.2)
    - Collect and input basic election information for the vote count algorithms to commence.
- Run an Instant Runoff Election with Given Ballots (Use case 4.3)
    - One of the main uses of the system is to run an instant runoff election and produce the results of that said run-off election.
- Run an OPL Election with Given Ballots (Use case 4.4)
    - Use Open Party Listing to calculate the results of an election
- Determine a Tie (Use case 4.5)
    - Randomly select the winner using a simulated coin flip or random number generator in the event of a tie.
- Create Audit File (Use case 4.6)
    - System creates an audit file for use that shows the winners, how the election progressed, and shows the overall election description such as type of voting, number of candidates, candidates, number of ballots. It will also show where the vote went for each person.
- Display Results to the User (Use case 4.7)
    - Display the winners of the election to the user once the results are determined
- Create Media File (Use case 4.8)
    - Create a summary file about the election that could be shared with the media.

2.3 User Classes and Characteristics

Typical user classes for this voting system include programmers, testers, and election officials. Programmers and testers have a technical background and will work on the software frequently to ensure accurate functionality. They will have full access privileges in the system in case any changes or bugs need to be fixed. Election officials likely will not have the technical background to work on the specifics of the program, but will be able to run the voting system to get the results of an election. Because of this, they will only have run privileges and will not be able to access the code. Election officials are expected to use this as frequently as there are elections.

2.4 Operating Environment

Software:

Must be able to have CSV files

Must run Java 8

Minimum System Requirements:

OS	RAM	Processor
Windows 10	16GB	Intel® Core™ i5 @ 3.4GHz (x4)
Mac OS Catalina 10.15	16GB	Intel Core i5 3.2 GHz
Ubuntu 20.04	16GB	Intel® Core™ i5 @ 800Hz (x3)

Recommended System:
OS: Ubuntu 20.04
RAM: 32GB Ram
CPU: Intel Core i7-4790 (Quad-Core) 3.60GHz

References:

https://cse.umn.edu/cseit/classrooms-labs

2.5 Design and Implementation Constraints

- The program has been written in Java 8 and therefore requires a compatible OS and command-line
- The program does not use any libraries not included in the standard java library.
- The program takes in either a CSV or CSV UTF 8 file that is within the same directory as input for the ballot information.
- Program should be able to run 100,000 ballots in under 8 minutes
- There are no specific safety or security requirements

2.6 User Documentation

Other than this document, there is no other documentation on how to operate the VCS . In section 4 of this document, we have laid out all possible use-cases for the system and in section 1.5 we have provided links to explanations of how our voting algorithm works.

2.7 Assumptions and Dependencies

- There are no errors in the CSV file or any ballots
- There are no write-in ballots
- The user knows how to use command line prompts
- The CSV file is stored in the same directory as the program
- There will always be only one file provided

3. External Interface Requirements

3.1 User Interfaces

The user interface for this system lives solely on the command line interface. When a user boots up the program they will be prompted to enter the name of the ballot file, which needs to be in CSV format and be in the same directory as the program. If the entered file does not exist, an error message will be provided and the user will be prompted again. The program will automatically produce an audit file and a media file both of which will be in a .txt format. The winners of the election will also be displayed on the terminal screen after the program has completed.

3.2 Hardware Interfaces

All user input is done with a keyboard, and all data is already in csv format in the designated directory. The data will be loaded and stored into system memory as it is read by the system. A summary file for the media, and an audit file will be written upon completion of the program.

3.3 Software Interfaces

The program was developed using Java 8, therefore Java 8 or better must be installed on the computer of choice.

3.4 Communications Interfaces

Since the program is run through the command line, all information will be displayed through the command line application. For Ubuntu and Mac operating systems this program should be run through the terminal. For Windows, the VCS should be run through the command line. All information will be displayed on either terminal or command line depending on the OS.

4. System Features

4.1: Bring a File into System

Name	Bring File into System
ID	FH_01
Description	Prompt the user for the name of the election file then open the file within the software
Actors	Election Official, Testers, System
Organizational Benefits	Ensures that the Ballot Data is available and ready to be processed by the voting algorithms.
Frequency of Use	Any time a new dataset of ballots wants to be analyzed. Should be used both for testing purposes and for an actual election cycle.
Triggers	Whenever a tester or election official boots up the program. The first prompt will ask the user what the name of the file is which stores the ballot information.
Preconditions	The file which is intended to be read in must have no numbering mistakes, be in the CSV format, and be in the same directory as the program.
Postconditions	The file has been loaded into the system and can now be processed to determine the outcome of an election.
Main Course	1. Once the user starts up the program, the system will prompt

	the user for the file name. 2. User types in the filename that contains the ballot in the format of “filename.csv” (EX1) 3. System ensures that the file exists and is accessible (EX 2) 4. System provides a confirmation message that the file has been read in correctly.
Alternate Courses	N/A
Exceptions	EX1 File Not in the correct format: 1. System provides user with error message that the file name they entered does not end with .csv 2. System prompts user again for the filename that contains the ballot in the format of “filename.csv” 3. If the format is correct, System continues at main course #3 4. If the format is not correct the system provides an error message that the file must be of type csv and that it should be inputted as “filename.csv” 5. Program exits EX2 File not Found: 1. System provides the user with an error message the file cannot be found in the directory and instructs them to double check the filename. 2. System prompts user again for the filename that contains the ballot in the format of “filename.csv” 3. System ensures that the format is correct (EX1) 4. If the file is found within the directory, the system continues to Main Course #4 5. If the file is not found the system provides the user with an error message saying the file is not in the directory. 6. Program exits

4.2: Preprocessing Data from Ballot File

Name	Preprocessing Data from Ballot File
ID	FH_02
Description	Collect and input basic election information for the vote count algorithms to commence.
Actors	System
Organizational Benefits	Ensures that the voting algorithms have all of the necessary

	information and data-structures to be able to run.
Frequency of Use	Any time a new dataset of ballots wants to be analyzed. Should be used both for testing purposes and for an actual election cycle.
Triggers	Immediately upon completion of FH_01.
Preconditions	The ballot file has been loaded into the system and can now be processed to determine the outcome of an election.
Postconditions	All data structures and attributes have been instantiated and initialized so the voting algorithms can run.
Main Course	1. System transfers all of the data within the file into a local 2D array to be processed. 2. Use the first line in the file to determine what type of voting algorithm will be used (IR or OPL) and save it within the system 3. Use the second line in the file to determine how many candidates are running in the election and save it within the system. 4. Using the third line and the information from Main Course#3, create individual data structures for each candidate (AC1) 5. Assign every ballot an individual, unique ID 6. Move every ballot into a candidates data structure based on their first choice (#1 rank)
Alternate Courses	AC1 The voting algorithm used is Open Party Listing: 1. Using the third line and the information from Main Course#3, determine the amount of parties and save that within the system 2. Using information from AC#1,build individual data structures for every political party 3. Using the third line, create individual data structures for each candidate within their party's data structure 4. Continue to Main Course#5
Exceptions	N/A

4.3: Run an Instant Runoff Election with Given Ballots

Name	Run an Instant Runoff Election with Given Ballots
ID	IR_01
Description	One of the main uses of the system is to run an instant runoff election given ballots stored in CSV and produce a result of that said run-off election.
Actors	System
Organizational Benefits	One of the main purposes of the system is to do an instant runoff election given ballots.
Frequency of Use	Every time the program runs with a valid CSV file containing ballot information, and the first line of the CSV is the string "IR".
Triggers	The program is run, the user gives the correct file when prompted by the program, and the first line of the CSV is the string "IR".
Preconditions	The program starts running, and a correctly formatted CSV file is read into the system. The first line of the CSV must be the string "IR".
Postconditions	A winner is determined and information is sent to the display winner part of the program.
Main Course	1. The CSV file is read in with the key string "IR" as 1st line 2. Information is stored while scanning the CSV file, and votes are tallied 3. If one candidate reaches majority of the vote, the winner declared as that candidate, and information will be sent to the display winner process and audit file 4. If there is not a candidate with a majority of votes (AC #1) 5. If there is only two candidates and there is a tie in votes or if the bottom two candidates are tied (AC #2) 6. Winner information is sent to display winner mechanism
Alternate Courses	AC #1 No Majority is reached at the end of a round 1. The candidate with the least number of votes is eliminated 2. AC #2, if there is a tie for last place 3. The votes are redistributed to other candidates based on their next choice 4. If no majority is reached, repeat to step 1 of AC #1, else MC step 3

	AC #2 There is a tie in voting with two candidates left or 1. The DT_01 (coin flip) is used 2. The winner of the coin flip is the of the round 3. AC #1 step 3, if there is two or more candidates left 4. MC step 6, if there are only two candidates left
Exceptions	N/A

4.4: Run an OPL Election with Given Ballots

Name	Run an OPL Election with Given Ballots
ID	OPL_01
Description	Use Open Party Listing to calculate the results of an election
Actors	System
Organizational Benefits	Allows the voter to vote for a candidate and a party with one vote. The popularity of the candidate within a party determines the ranking within that party.
Frequency of Use	Every time Open Party Listing must be used to determine an election
Triggers	Whenever the csv file indicates that the program must use open party listing to determine the election
Preconditions	File must be brought into the system with the OPL Election Type
Postconditions	Seats are determined based on the winning parties and candidates within the party, following the largest remainder approach to seat allocation
Main Course	1. CSV File is read in with the OPL Election Type indicated 2. The voting information is stored as the file is read in 3. All independent candidates are grouped into one party 4. The order of candidates within parties is determined based on which candidate receives the most votes 5. Seats are given to the parties based on what percentage of the vote the parties receive 6. The remainder of the votes are determined with the largest remainder approach

Alternate Courses	If there is a tie between candidates in a party: 1. See DT_01 to determine which candidate gets a higher seed within the party If there is a tie between parties 1. See DT_01 to determine which party gets the seat
Exceptions	N/A

4.5: Determine a Tie

Name	Determine a Tie
ID	DT_01
Description	Randomly select the winner using a simulated coin flip or random number generator in the event of a tie.
Actors	System
Organizational Benefits	Ensures that the voting system will always determine a winner of the election even when there is a tie. By being random it will also ensure that election was fair.
Frequency of Use	Only in the event that two or more people are tied as winners which should be very rare depending on the size of the election.
Triggers	Whenever a determination needs to be made and two or more candidates or parties have the same number of votes. For example, this could be used when two or more candidates both have the lowest number of votes and one of them needs to be redistributed. In Open Party List this could be when two or more parties have equal remaining votes to distribute the remaining seats.
Preconditions	A decision needs to be made between two or more parties to continue the voting algorithm and they have an equal number of votes.
Postconditions	A decision has been made in a fair way and the algorithm can continue on.
Main Course	1. When running through the algorithm, a comparison needs to be made between two candidates or parties but they have the same number of votes.

	2. Assign head and tails to the parties in question 3. Use the random library functions to simulate a coin flip 4. Record winner in audit file and continue the algorithm.
Alternate Courses	1. More than two candidates or parties are tied and a distinction needs to be made 2. Assign numbers 1 through n to the parties where n is the number of tied candidates or parties 3. Use the random library functions to select a number from 1 to n 4. The selected number and associated candidate or party is then dealt with.
Exceptions	N/A

4.6: Create Audit File

Name	Create Audit File
ID	AUF_01
Description	System creates an audit file for use that shows the winners, how the election progressed, and shows the overall election description such as type of voting, number of candidates, candidates, number of ballots. It will also show where the vote went for each person.
Actors	System
Organizational Benefits	The creation of an audit file allows for recorded detailed information about the system processes and the results of the election. The audit file can be used to verify the accuracy of the overall system.
Frequency of Use	Anytime an election file is taken by the system and the results are calculated.
Triggers	When election results are being calculated, all the processes will be written onto the audit file, the file will be finished by end of the program.
Preconditions	A CSV file containing all election data must be read into the system properly. Program and vote counting has to start.
Postconditions	The audit file will be saved in txt form with the name

	"audit_year_month_day_hour_seconds" where the year_month_day_hour_seconds is the year, month, day, hours, sec that the program is run. File will be saved in the same directory as the program.
Main Course	1. CSV file is created with name audit_year_month_day_hour_seconds 2. Once the first round of voting is counted by the system, the data of vote counts is written to the audit file 3. Data is added to the audit file between each round of voting counted 4. Once the votes are counted by the system, general information is added to the audit file. 5. The audit file is saved and is stored in the same directory as the program
Alternate Courses	N/A
Exceptions	N/A

4.7: Display Winners and Information

Name	Display Winners and Information
ID	DW_01
Description	Display the winners of the election to the user once the results are determined
Actors	System, User
Organizational Benefits	Displaying the results in an easily readable format for the user simplifies the system user interaction, decreasing the chance for any user mistakes.
Frequency of Use	Every time the program finishes calculating an election, the results are shown.
Triggers	Whenever the program is done reading through an election file, the outcome of that election is displayed.
Preconditions	An election needs to be calculated, and the results need to be determined by the program

Postconditions	Results of the election are displayed to the user in an easily readable format via the terminal
Main Course	1. After running the algorithm, the key values to be displayed need to be identified 2. The values are formatted into a table 3. The table is displayed in the console
Alternate Courses	N/A
Exceptions	N/A

4.8: Create Media File

Name	Create Media File
ID	MF_01
Description	Create a summary file about the election that could be shared with the media.
Actors	System
Organizational Benefits	This file will be sent to the media so that they can inform the public about the election results.
Frequency of Use	Everytime the software is run on a data file for an election or for testing.
Triggers	When the voting algorithm is done and the relevant information about the election has been computed and organized.
Preconditions	The election algorithm is done and there are calculated statistics stored regarding candidates, vote count for each, type of election, and total votes.
Postconditions	File is created in the same directory as the program that summarizes the election and is intended for media release.
Main Course	1. Gather information regarding candidates, vote count for each, type of election, and total votes. 2. Format information into a reader-friendly text file 3. Save file in the same directory with name election_media_~DATE~.txt where ~DATE~ is the unique time the election count was run at.

Alternate Courses	N/A
Exceptions	N/A

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The VCS should be able to run 100,000 ballots in under 8 minutes for either type of voting algorithm. This is to ensure that the software can deliver quick results and could scale for larger voter turnout elections.

5.2 Safety Requirements

There are no specific safety requirements.

5.3 Security Requirements

There are no specific security requirements.

5.4 Software Quality Attributes

The VCS must be able to calculate election results for both Open Party Listing and Instant Runoff elections. This is so that the same software can be used in different areas where one or the other voting system might be used. This also means that the software is adaptable in the event that a location changes the type of voting algorithm used. The software correctly identifies a winner to an OPL or IR election given that the provided data is correct. In the event of a tie, it will simulate a coin flip or use a random number generator to determine a winner. The VCS is structured in a way such that additional voting algorithms can be easily added to support more voting systems in the future. This software is also structured so that testers can verify and maintain the validity of the results. This software only requires a basic knowledge of running programs on the command line to operate.

5.5 Business Rules

Election officials can run the program with an existing csv data file at any time and access the produced results. System testers and programmers can also do this and additionally have the ability to access the source code and make adjustments as needed.

Appendix A: Glossary

In this document, IR refers to Instant Runoff and OPL refers to Open-Party Listing voting. Along with this, CSV refers to a .csv file or comma separated value file. VCS refers to our software's name which stands for Voting Count System.