
Software Requirements Specification

for

Voting System

Version 1.0

Sean Beaulieu- beaul116

Ann Huynh - huynh441

Shey Pemberton - pembe020

Jasper Rutherford - ruthe124

University of Minnesota

CSCI 5801

February 2022

Table of Contents

Table of Contents.....	1
Revision History.....	2
1. Introduction.....	2
1.1 Purpose.....	2
1.2 Document Conventions.....	2
1.3 Intended Audience and Reading Suggestions.....	2
1.4 Product Scope.....	3
1.5 References.....	3
2. Overall Description.....	3
2.1 Product Perspective.....	3
2.2 Product Functions.....	3
2.3 User Classes and Characteristics.....	4
2.4 Operating Environment.....	4
2.5 Design and Implementation Constraints.....	4
2.6 User Documentation.....	4
2.7 Assumptions and Dependencies.....	5
3. External Interface Requirements.....	5
3.1 User Interfaces.....	5
3.2 Hardware Interfaces.....	5
3.3 Software Interfaces.....	5
3.4 Communications Interfaces.....	5
4. System Features.....	5
4.1 Read File Name.....	5
4.2 Shuffle Ballots.....	6
4.3 Run Instant Runoff Voting.....	7
4.4 Run Open Party Listing.....	8
4.5 Flip a Coin.....	10
4.6 Display Winner to Screen.....	11
4.7 Produce Audit File.....	12
4.8 Produce Report.....	13
5. Other Nonfunctional Requirements.....	13
5.1 Performance Requirements.....	13
5.2 Safety Requirements.....	13
5.3 Security Requirements.....	14
5.4 Software Quality Attributes.....	14
5.5 Business Rules.....	14
Appendix A: Glossary.....	14
Appendix B: Analysis Models.....	14
Appendix C: Examples.....	14

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to describe the software which will provide a voting system. This product will allow for two types of voting: Instant Runoff Voting and Open Party Listing. The software will allow the user to decide which of the two voting systems to use. The document will describe the features of the software, and how the software will be used to calculate the winning candidates. Along with the software uses, this document will also detail the constraints of the system. This is version 1.0.

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

This document is intended to be read by election officials, programmers, testers, and media personnel. The following sections of this SRS include an overall description of the product, external interface requirements, system features, nonfunctional requirements, and appendices including a glossary.

All readers should start with the overall description following this introduction, and refer to the glossary for any jargon or abbreviations as needed.

- *Programmers and testers* should read this document in full, and may find sections 4.2, 4.5, and 4.7 particularly helpful.
- *Election Officials* should focus on sections 4.1, 4.3, 4.4, and 4.6
- *Media personnel* should focus on sections 4.6 and 4.8

1.4 Product Scope

This software program will consist of a voting system that uses two types of voting systems: Instant Runoff Voting and Party List Voting using Open Party List. The benefits of this software will be that the program can process ballot data intended for either of the described voting algorithms, and the program can be used multiple times a year for local and general elections. This versatility allows the program to be useful for any type of election. The long-term goal is for this system to become part of an integrated online voting system, in which the preprocessing will be done by the system described in this document. Please refer to the Project 1 Waterfall Methodology SRS Document for Voting System for further details, provided in the References below this section.

1.5 References

[IEEE Template for System Requirement Specification Documents](#)

Project 1 Waterfall Methodology SRS Document for Voting System

Provided by Shana Watters for the 5801 Software Engineering course at UMN

Use Case Structured Specification Template Example

Provided by Shana Watters for the 5801 Software Engineering course at UMN

2. Overall Description

2.1 Product Perspective

This product is an implementation of existing voting algorithms, Instant Runoff Voting and Party List Voting with Open Parties, that are contained in one program with the ability to process ballots designed for either voting algorithms. This is a new, self-contained product; the long-term goal is for this product's voting system to be integrated into an online voting system.

2.2 Product Functions

For the context of this Product Functions section, assume "user" refers to an election official operating the program for an official election unless otherwise specified.

- The user enters the name of a CSV file containing ballot data into the command line.
- *For programmers and testers only:* the user may enable or disable shuffling of ballots by entering the appropriate command into the command line.
- System reads in ballots from input file and processes them according to election type.
- System produces an audit file.
- System produces a report file.
- System will display results of the election on the screen.

For a visual, please refer to Appendix B.

2.3 User Classes and Characteristics

The primary user class for this product is election officials. The only users who will use this product in a functional sense are election officials. The other user classes who may interact with this product are programmers, testers, and media personnel. All user classes should have proper security clearance to interact with the product or run official elections, but privileges are not determined by this product (see section 5.3 Security Requirements for further explanation).

- *Programmers and testers:*
 - Will use this system to test and maintain on a demand basis
 - Will ideally have the most technical expertise out of any user class
- *Election Officials:*
 - Will use this system whenever an election needs to be calculated
 - Will be assumed to have limited technical expertise outside of system scope and manual; it is important they are familiar with system features
- *Media personnel:*
 - May only receive results of official elections run

2.4 Operating Environment

The operating environment for this system will be Linux, using up-to-date CSELabs machines.

2.5 Design and Implementation Constraints

- The runtime constraint is 100,000 ballots in under 8 minutes
- The program is written in Java and runs through command prompt
- Votes must be able to be counted through scanning of a CSV file
- The file is exported from Excel into CSV formatting
- The user must be able to input information through the command line.
- The program must handle both IRV and OPL election types
- The election file will be located in the same directory as the program
- Program will be run multiple times during the year at normal election times and special elections

2.6 User Documentation

A user manual will be delivered along with the software that will explain how to operate the program. This manual is intended to be used by election officials.

2.7 Assumptions and Dependencies

Assumption that the program is run on a linux OS, CSElab machine. Assumption that the election file is formatted correctly (a single CSV file, first line indicates type of voting, exported from excel, as specified in the Project 1 Waterfall Methodology SRS Voting System document). Dependency that the election file is located in the same directory as the program. Dependency that software interfaces are updated to code so the system can run Java programs. Assumption that security such as privileges are determined by election official organizations instead of this program.

3. External Interface Requirements

3.1 User Interfaces

Command line interface. If there are user errors, errors will be displayed in the terminal.

3.2 Hardware Interfaces

Direct hardware interfaces are not necessary for this system.

3.3 Software Interfaces

Linux, Java, jdk version, all the particular details, input file of the ballots, output on command line of results.

The software will be run on a Linux operating system, and be coded in Java.

3.4 Communications Interfaces

Requires an external input file. Generates report files that may be communicated to external teams. System side communications are not necessary for this type of software.

4. System Features

4.1 Read File Name

Name	Input Election File Name
------	--------------------------

ID	001
Description	Users can submit the name of the file to read ballots in from.
Actors	Election Officials, testers, programmers
Organizational Benefits	Allows for flexibility with file inputs/names
Frequency of Use	Supposed to be used only once per election; only used multiple times to correct user input errors.
Triggers	Command is run from the command line
Preconditions	Provided input file should follow the correct CSV formatting (as specified in the Project 1 Waterfall Methodology SRS Document for Voting System, see References) and be located in the same directory as the program.
Postconditions	The filename for ballot reading is set equal to the filename that the user inputs.
Main Course	<ol style="list-style-type: none"> 1. User inputs command to run with the desired filename as an argument 2. Filename is set equal to supplied argument
Alternate Courses	<ol style="list-style-type: none"> 1. User inputs the wrong filename 2. User uses the command again to input a correct filename.
Exceptions	During this use case, the system does not check for correct file contents so there are no exceptions.

4.2 Shuffle Ballots

Name	Shuffle or not shuffle ballots
ID	002
Description	Used to determine if the system will shuffle the ballots before calculating IRV elections.

	(If not used, shuffle defaults to on)
Actors	Programmers, testers
Organizational Benefits	Allows for the user to test if the program determines the correct winner based on the exact order of ballots.
Frequency of Use	Once per election
Triggers	The shuffle command is entered
Preconditions	IRV election has been chosen
Postconditions	Sets the ballots to be shuffled or not shuffled in following algorithm
Main Course	1. User will enter the shuffle command with the argument 1 for shuffle and 0 for no shuffle 2. Set indicator according to user input
Alternate Courses	N/A
Exceptions	If the ballots are to be counted for an open party listing, the user cannot set the ballots to be shuffled.

4.3 Run Instant Runoff Voting

Name	Run Instant Runoff Voting
ID	003
Description	Calculate winners based on majority preferential voting.
Actors	Election officials and testers
Organizational Benefits	Improvement over two-round system: “instant” runoff
Frequency of Use	Once per election
Triggers	They run the command in the command line
Preconditions	<ul style="list-style-type: none"> - CSV file name has been read in - CSV file needs to be located in the

	same directory as the program - Shuffle must be enabled, as IRV ballots must be in random order to avoid preference for early voters
Postconditions	Calculate winners based on fair instant runoff voting election type. Winners are determined by the ranking of candidates, then winners are declared onscreen in a general report, and an audit report is created from results.
Main Course	1. Run the relevant command in the command line to receive input file 2. Shuffle the ballots 3. Begin counting first choice on ballots and continually test for winner until first place is determined 4. Continue counting second choice on following ballots without shuffling and continually test for winner until second place is determined 5. Results displayed
Alternate Courses	In the case of a tie for any ranking of candidate, use a coin flip to determine the winner If there is not a clear majority, popularity wins after all votes have been handed out
Exceptions	The IRV command is run, but the election has already been calculated so the screen displays an error message in response and gives the option to run again

4.4 Run Open Party Listing

Name	Run Open Party Listing
ID	004
Description	Calculate winners based on the open party listing election type. Winners are determined

	by the number of seats and popularity of the candidates, using the largest remainder formula
Actors	Programmers, testers, election officials
Organizational Benefits	Improvement over two-round system: “instant” runoff. Party list voting is the most popular form of proportional representation and is used worldwide.
Frequency of Use	Once per election
Triggers	User inputs the command for running an open party listing election:
Preconditions	<ul style="list-style-type: none"> - CSV file name has been read in - CSV file needs to be located in the same directory as the program - Order of candidates under a given party does not need to have any particular order, so no shuffle should be selected - Number of parties and candidates have been determined
Postconditions	Winners are determined from a fair open list party election and then winners are declared in a general report on screen. An audit report is created from results. An example of the general report can be seen in Appendix C.
Main Course	<ol style="list-style-type: none"> 1. Pass through the ballots and determine the total number of votes, and the votes each party receives. 2. Divide the number of votes by the number of seats to calculate the quota. 3. Divide the votes each party receives by the quota, and the party wins one seat for each whole number produced. 4. The remainder votes for the parties are compared, and the parties with the largest remainders are allocated the remaining seats.
Alternate Courses	In the case of a tie for any ranking of candidate, use the droop quota and then a coin flip to determine the winner

Exceptions	<p>1. The OPL command is run, but the file information is not formatted for an OPL election. The program displays an error message to the screen and gives the user an option to run again from the start of the program.</p> <p>2. The OPL command is run, but the user has previously indicated the ballots should be shuffled. The program displays an error message to the screen and gives the user the option to run again from the start of the program.</p> <p>2. The OPL command is run, but the election has already been calculated so the screen displays an error message to the screen and gives the option to run again from the start of the program.</p>
-------------------	---

4.5 Flip a Coin

Name	Coin flipping
ID	005
Description	In the event of a tie, a coin is flipped to determine the winner.
Actors	Election officials and testers
Organizational Benefits	Provides a way to determine winner in event of a tie
Frequency of Use	Any time a tie is determined
Triggers	The voting system is run and a tie is the pre-finished result.
Preconditions	The number of votes for one candidate is equal to the number of votes for another.
Postconditions	One candidate is declared the winner of that vote.
Main Course	1. System is run and votes are calculated

	<ol style="list-style-type: none"> 2. An even tie is determined to be the result 3. A coin flip will be run to determine the winner 4. A winner is determined
Alternate Courses	A tie is not the end result of tallying votes, and a coin flip is not necessary
Exceptions	N/A

4.6 Display Winner To Screen

Name	Display Winner To Screen
ID	006
Description	Upon completion of vote tallying, display the winner(s)
Actors	Election officials, testers
Organizational Benefits	Easy to identify the results of the election
Frequency of Use	Once per cycle of votes
Triggers	They run the software on a list of votes
Preconditions	System has processed votes and determined a winner
Postconditions	Winner(s) are determined from a fair election, then winner(s) are declared onscreen.
Main Course	<ol style="list-style-type: none"> 1. Run the relevant command in the command line 2. Determine winners and losers 3. Results displayed
Alternate Courses	In the case of a tie for any ranking of candidate, use a coin flip to determine the winner
Exceptions	The IRV command is run, but the election has already been calculated so the screen displays

	an error message in response and gives the option to run again
--	--

4.7 Produce Audit File

Name	Produce Audit File
ID	007
Description	An audit file is produced with the election information
Actors	Testers, Programmers
Organizational Benefits	All election information (type of voting, number of ballots, etc) are all stored in one place
Frequency of Use	Once per election
Triggers	IRV/OPL system is being run
Preconditions	Election should be run
Postconditions	Audit file does not have to be shown unless requested
Main Course	<ol style="list-style-type: none"> 1. Input file is given 2. System runs ballots 3. Audit file keeps track of how election has progressed and all the steps 4. Winners are listed 6. Audit file stores all the information
Alternate Courses	If IRV, the audit file contains who received what ballot, the order of ballots, the order of removal of candidates, and what ballots were redistributed.
Exceptions	If an audit file is already produced for the election, you would not produce another one

4.8 Produce a Report

Name	Produce a Report
ID	008
Description	A report with statistics of results (what winners had how many votes, etc)
Actors	Media Personnel
Organizational Benefits	Specific information needed for reporting is all in one place
Frequency of Use	Once per election
Triggers	IRV/OPL System has completed run
Preconditions	Election should be run
Postconditions	Report created for viewing will be stored in the same directory as the program
Main Course	<ol style="list-style-type: none">1. Input file is given2. System runs ballots3. System displays winners and losers4. Report file created with statistics of the results
Alternate Courses	
Exceptions	If a report file is already made, another one would not be produced

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The program's runtime must be under 8 minutes for 100,000 ballots.

5.2 Safety Requirements

There are no safety requirements for this program, because security for ensuring legal votes is handled at voting centers before the file for this program is generated.

5.3 Security Requirements

No checks are run in the program to ensure any level of security around the ballots. It is assumed that secure ballot information is delivered to the system.

To ensure that all votes are tallied fairly, an audit file can be generated to see exactly how the votes were counted.

5.4 Software Quality Attributes

Ease of use for Election Officials who run the program, as the program is intended to take in input files so the officials do not have to enter large amounts of information while running the program. Flexibility as there are two different voting algorithms that can be applied to ballot data. Adaptability in the case of ties which will automatically be decided with a random coin flip. Reusability, so the program can be run for multiple elections in a year.

5.5 Business Rules

Programmers and Testers will be running and using this program in full during development. Once the program is functional, only individuals in the role of Election Officials can operate the program with the command line, but no special permissions are defined within the program as it is assumed that security will be handled by the organizations which employ the program. Media Personnel can have the results of the election shared with them.

Appendix A: Glossary

CSELabs: College of Science and Engineering computer labs at UMN

CSV: Comma Separated Value - A type of file that contains values separated by commas.

IEEE: Institute of Electrical and Electronics Engineers: The institute that made the standards this document is written by.

IRV: Instant Runoff Voting - A type of voting system where voters 'rank' their candidates instead of choosing a single candidate. Voters may rank the candidates from their first choice to the total number of candidates.

- The ballots are shuffled to ensure that early votes do not take priority.
- When the ballots are processed, on the first pass through, only the first choice candidates are counted and if a candidate receives over 50% of the votes that candidate is declared the winner. The remaining ballots are processed in the manner described below.
- If no candidate receives a majority, then the candidate with the fewest votes is in last place. The ballots of supporters of this last place candidate are transferred to whichever

of the remaining candidates they marked as their second choice. After this transfer, the votes are recounted (without shuffling) to see if any remaining candidate receives a majority of the vote.

- The process of eliminating the least favorable candidate and transferring their votes continues until one candidate is the majority and wins the election.

OPL: Open Party Listing - A type of voting system where voters pick a single political party who has put up candidates to take over seats, which are then distributed proportionally by the percentage of votes each party receives. The ballot contains an unordered list of candidates chosen in primaries; voters cannot choose parties directly, and instead vote for individuals, which count for their party. Independents are grouped into one party. Seat allocation will be determined by the largest remainder formula as described below.

- The ballots are processed to determine the total number of valid votes in the district, and to determine the number of votes that each party receives.
- The total number of votes are divided by the number of seats to calculate the quota.
- In the first allocation of seats, the total votes for each party is divided by the quota, and each party wins one seat for the whole number produced by this division.
- The second allocation depends on the remainder of these divisions, where the remainders are compared and the parties with the largest remainders are allocated the remaining seats.

SRS: Software Requirements Specification - A software document that details all of the requirements and information about the system it is associated with.

UMN: University of Minnesota Twin-Cities

Appendix B: Analysis Models

For an initial visual containing the order of use cases and notes on system requirements, please visit [this link](#) to view a Mural collage.

Appendix C: Examples

Example of An Open Party Listing election general report as shown in the Project 1 Waterfall Voting SRS Voting System reference document.

Parties	Votes	First Allocation Of Seats	Remaining Votes	Second Allocation of Seats	Final Seat Total	% of Vote to % of Seats
Republican	38,000	3	8,000	1	4	38% / 40%
Democratic	23,000	2	3,000	0	2	23% / 20%
Reform	21,000	2	1,000	0	2	21% / 20%
Green	12,000	1	2,000	0	1	12% / 10%
Moll	6,000	0	6,000	1	1	6% / 10%