
Software Requirements Specification

for

Voting Tabulation System

Version 1.0

Prepared by Brian Bianchi, Andrew Brevick, Cuong Ha

University of Minnesota Twin Cities - Spring 2023 - Group 21

02/16/2023

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	3
2.4 Operating Environment	3
2.5 Design and Implementation Constraints	3
2.6 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	5
3.3 Software Interfaces	6
3.4 Communications Interfaces	6
Instant Runoff Voting	6
Closed Party List Voting	7
4. System Features	8
4.1 Use Cases	8
4.2 Results File	9
4.3 Audit File	9
5. Other Nonfunctional Requirements	10
5.1 Performance Requirements	10
5.2 Safety Requirements	10
5.3 Security Requirements	11
5.4 Software Quality Attributes	11
5.5 Business Rules	11
6. Other Requirements	11
Appendix A: Glossary	11
Appendix B: Analysis Models	11
Appendix C: To Be Determined List	12

Revision History

Name	Date	Reason For Changes	Version
Andrew, Brian, Cuong	February 16, 2023	Initial revision	1.00

1. Introduction

1.1 Purpose

- This voting system:
 - Provide a secure and efficient platform for conducting elections, which will be used multiple times per year for standard and special elections.
 - Support two voting methods to cater to the specific requirements of different election processes:
 - Instant Runoff (IR): allows voters to rank their preferred candidates in order of preference.
 - Close Party List (CPL): allows voters to choose a political party and have their vote count towards the party's overall tally.
- This SRS document:
 - Present a detailed description of the voting system.
 - Explain the purpose and features of the software, the primary modes of user interaction, and proposed formats for the user interface and outputs.
 - Intended for both development staff and users of the system.
 - Changes to the document's content will be recorded.

1.2 Document Conventions

- Document style: IEEE System Requirements Specification 1999 Template
- Font: Time News Roman
- Font sizes:
 - Heading 1: 18
 - Heading 2: 14
 - Normal text: 11

1.3 Intended Audience and Reading Suggestions

- Programmers / Testers / Project Managers - Members of the initial development team or those modifying the software in the future
- Election officials - The people who use the system to tally election results
- Auditors - The people who verify the accuracy of election results

1.4 Product Scope

- This system is an electronic tool to tabulate ballots and to determine the winner(s) in a few common types of election.
- A CSV file containing the election results will be given to the system by the user.
- The system reads and uses the provided data to tabulate and finalize the elections.
- The methodology of how the calculations are done depends on the chosen election type.
- The process is automated to substantially reduce the potential for user error and to decrease the amount of time required for tabulation.

1.5 References

- Learn more about Instant Runoff Voting and Closed Party List
 - Title: Project 1 – Waterfall Methodology
 - Author(s): Shana Watters
 - Organization: University of Minnesota - Twin Cities
 - Publication: Spring 2023
 - Source: canvas.umn.edu/courses/355321/files/33732199?module_item_id=9692478
- More information about Instant Runoff Voting
 - Title: Ranked Choice Voting
 - Source: fairvote.org/our-reforms/ranked-choice-voting/
- SRS Template can be downloaded here
 - Title: Software Requirements Specification Template
 - Author(s): Karl E. Wiegiers
 - Organization: IEEE Computer Society
 - Publication: 1999
 - Source: canvas.umn.edu/courses/355321/files/33732268?module_item_id=9692479
- More information about GCC Compiler
 - Source: gcc.gnu.org/

2. Overall Description

2.1 Product Perspective

- This system is a standalone software program designed to be used by election officials to tabulate ballots and determine the outcomes of elections.
- This system will have no direct interactions with other software systems.
- This system operates on pre-collected ballots for two election types: Instant Runoff (IR) and Closed Party List (CPL).
- This system is designed to be operated by election officials with limited knowledge of the software itself.

2.2 Product Functions

- Load ballot file
- Read ballot file header
- Process election
- Break a tie
- Display result page
- Create audit file
- Tabulate IR
- Tabulate PCL
- Name audit file
- Resolve inconclusive IR as popularity vote
- Extract additional information from the user
- Read ballot file

2.3 User Classes and Characteristics

- Internal users:
 - Election officials are the primary intended users of this system.
 - May not be knowledgeable in the operation of the program.
 - The only purpose is to obtain the election results.
 - Must be able to perform simple interactions with the Linux OS.
 - Must be familiar with standard GUI interactions, such as buttons and field entry.
 - Programmers and testers will be less frequent users of the system.
 - Must be Linux literate.
 - Must be able to operate programs via command line inputs.
 - No user type will have special privileges or permissions.
- External users:
 - The election results may be shared with media personnel, but they will not interact with the program directly.

2.4 Operating Environment

- The system should perform well on Ubuntu 20.04.1.
- This specific operating environment will be utilized during the development process, allowing for thorough testing and optimization of the system's performance.

2.5 Design and Implementation Constraints

- Required programming language(s): C/C++.

2.6 Assumptions and Dependencies

- The system is developed entirely in C/C++, and therefore requires a corresponding compiler to be installed on the user's machine. A recommended compiler is GCC 9.4.0.
- Source files and make files for the system will be provided.
- The system should be able to be initiated from the command prompt.

3. External Interface Requirements

3.1 User Interfaces

- There will be two “modes” for the software:
 - Standard mode: contains a GUI made using GTK, and is accessible for all users.
 - Maintenance mode: contains exclusively command line interface, and is manually accessible by the programmers and testers on launch with command line arguments.
- The interface for both will need to provide visual output and at designated times take user input.
- Any windows along the way generated for the GUI will have standard OS controls (minimize and close) which will be handled appropriately.
- The command line interface will provide similar functionality to the GUI, but any displayed content will be plaintext, and any inputs will be entered via the command line.
 - File path entry becomes a command line entry.

- Confirmation buttons become (y)es/(n)o text entries.

Details of the GUI:

1. Once the system is launched, there will be two text entry boxes for election date (in mm/dd/yyyy format) and file name. Entering the file path to the desired file and clicking “Confirm” (shortcut *c*) will bring the user to the next interface.

The screenshot shows a window titled "Election Tabulation Software" with a close button (X) in the top right corner. The main heading is "Election File Selection". Below the heading is a paragraph of instructions: "Please enter date of the election (in mm/dd/yyyy format) in the top box and then enter the name of the election results file to load. This file should be in the same folder as this application. Please include any file extensions at the end of the name (.txt, .csv, and so on).". There are two text input fields: the first is labeled "Enter Election Date Here (mm/dd/yyyy)..." and the second is labeled "Enter File Name Here...". At the bottom is a green button labeled "Confirm".

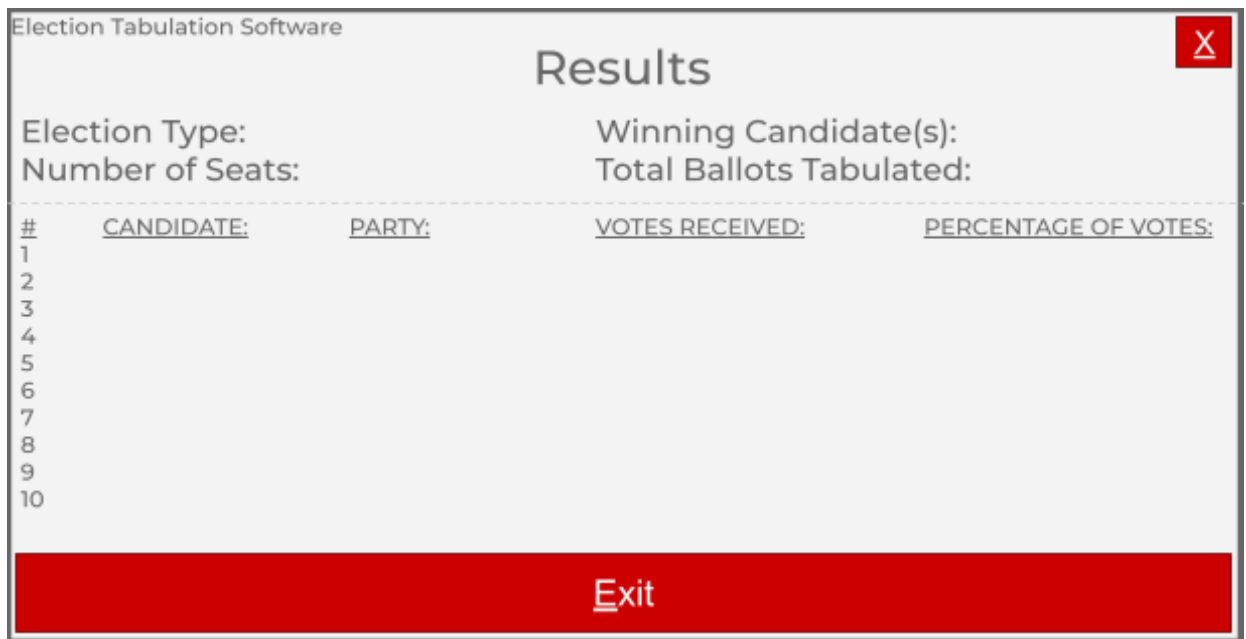
2. After the user enters and confirms the file path, a small information text box containing the header information of the file will be shown to the user, or an error message will be displayed if something is wrong. The user can confirm the validity of the file (shortcut *C*) or revert back to the previous interface (shortcut *B*) if they opened the wrong file. If confirmed, the system will proceed to the next step and direct the user to the next interface.

The screenshot shows a window titled "Election Tabulation Software" with a close button (X) in the top right corner. The main heading is "Election Information". Below the heading is the following text: "Election Type: Instant Runoff", "Number of Candidates: 4", "Candidates: Rosen (D), Kleinberg (R), Chou (I), Royce (L)", and "Ballot Quantity: 6". At the bottom are two buttons: a yellow button labeled "Back to File Selection" and a green button labeled "Confirm".

3. In this interface, the system will notify the user that it is tabulating the ballots.



4. When tabulation is complete, it will display a results screen showing certain information as text to the user.



3.2 Hardware Interfaces

- The hardware accessed in question will be a CSE lab machine.
- The system will be displayed on a connected monitor.
- A keyboard is required, as it is used to interact with all core functionality of the system.
- A mouse is optional, though the system will still support the use of it in standard user mode.

3.3 Software Interfaces

The software will be primarily written in C/C++ on the CSE lab machines, which at time of documentation run on Ubuntu 20.04 and feature GCC for compilation of C++ software. The CSE Lab machines natively utilize the Gnome desktop environment, which our interface toolkit GTK supports.

3.4 Communications Interfaces

- No external connection is required to operate for this release of this system.
- Missing information from the file will be prompted to the user to enter in the GUI.

Information about election type and candidates will be extracted from the header of the input file(s). Note that at this stage, only one file containing ballot information will be input to the system. The following sections summarize the procedure for two types of elections, Instant Runoff (IR) and Closed Party List (CPL), as defined at the time of writing. The header formats of the input files are reported. No write it candidates will be allowed.

Instant Runoff Voting

In this type of election, voters rank candidates in order of preference, instead of selecting only one candidate. A sample ballot is shown below.

Instant Runoff Ballot

Official Ballot Municipal Elections					
INSTRUCTIONS TO VOTERS Mark Your Choices by Filling in the Numbered Boxes Only Fill in the number one <input type="text"/> box next to your first choice; fill in the number two <input type="text"/> box next to your second choice; fill in the number three <input type="text"/> box next to your third choice, and so on. You may fill in as many choices as you please. Fill in no more than one box per candidate. Fill in no more than one box per column.	Candidates for City Council District One		<i>Only one vote per candidate</i> <i>Only one vote per column</i>		
	Susan Rosen (Democrat)		<input type="text"/>	<input type="text"/>	<input type="text"/>
	Nina Kleinberg (Republican)		<input type="text"/>	<input type="text"/>	<input type="text"/>
	Thomas Chou (Independent)		<input type="text"/>	<input type="text"/>	<input type="text"/>
	Edward Royce (Libertarian)		<input type="text"/>	<input type="text"/>	<input type="text"/>
	Write-In		<input type="text"/>	<input type="text"/>	<input type="text"/>
			<input type="text"/>	<input type="text"/>	<input type="text"/>
To Vote for a Write-In Candidate: Next to the name you have written in, mark a numbered box to indicate your choice of number for that candidate.					
Do Not Use Red To Mark Ballot					

Ballot input files will not deviate from the format defined below.

Example File Generated for Instant Runoff voting:

```
IR
4
Rosen (D), Kleinberg (R), Chou (I), Royce (L)
6
1,3,4,2
1,,2,
1,2,3,
3,2,1,4
,,1,2
,,,1
```

For this voting type, the header contains four lines, defined as follows:

1. Election Type
2. Number of candidates
3. Candidate names and affiliations, comma separated
4. Number of ballots

Each sequential line contains the vote information for a single ballot.

$p_1, p_2, p_3, p_4, p_5, p_6$

The line will contain integers at the positions of the candidates in the list on line 3. Each ballot will have a vote for at least one first choice candidate, marked with a 1. Ballots can rank any number of candidates up to the total number in the election. Candidates not ranked will have a blank entry in the comma separated line. Rankings will be assigned sequentially. In other words, there can be no third choice candidate without a second choice, and so on.

Closed Party List Voting

In this type of election, voters express a preference for a particular party instead of individual candidates. A sample ballot is shown below.

Closed Party List Ballot

Official Ballot Election for the United States House of Representatives District One				
Voting Instructions 1. You only have ONE vote. 2. Place an X in the box UNDER the party for whom you wish to vote.				
Democratic	Republican	Reform	Green	Independent Candidate
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1. Benjamin Foster	1. Wendy Berg	1. Steven Wong	1. Tom Wartenberg	1. Robert Moll
2. Sam Rosen-Amy	2. Steve Groinic	2. Deborah Gorlin	2. Juan Hernandez	
3. Colin Volz	3. Sarah McClurg	3. Brad Crenshaw	3. Beata Panagopoulos	
4. Benjamin Pike	4. Gerald Epstein	4. Daniel Czitrom	4. Alice Morey	
5. Megan Gentzler	5. Fran Deutsch	5. Meryl Fingrud	5. Sarah Pringle	

Ballot input files will not deviate from the format defined below.

```
CPL
6
Democratic, Republican, New Wave, Reform, Green, Independent
Foster, Volz, Pike
Green, Xu, Wang
Jacks, Rosen
McClure, Berg
Zheng, Melvin
Peters
3
9
1,,,,
1,,,,
,1,,,
,,,1,
,,,1
,,,1,,
,,,1,,
1,,,,
,1,,,,
```

For this voting type, the header contains a variable number of lines, defined as follows:

1. Election Type
2. Number of parties
3. Parties
4. Candidates for Party 1
5. Candidates for Party 2
6. ...
7. Candidates for Party N
8. Party names, comma separated
9. Number of seats up for election
10. Number of ballots

Independent candidates will be listed as a separate party, with a unique name (I.e. Independent 1, Independent 2). Also, since this is a closed party listing, the order of seats assigned will be the same as the order that candidates are listed within their party.

Each following line contains the vote information for a single ballot.

$p_1, p_2, p_3, p_4, p_5, p_6$

One position in the comma separated line will contain a 1, indicating a vote for the candidate at that position in the list on line 3. The remaining positions will be empty.

4. System Features

4.1 Use Cases

Refer to the *Use Cases* document for a comprehensive set of use cases for the system. It includes all interactions a user can have with the system, as well as the logical flow and conditional states associated with those interactions.

4.2 Results File

A results file will be generated containing the same information that is presented on the results screen. The file name will contain the same information as the audit file.

4.3 Audit File

An audit file will be generated which includes type of voting, number of candidates, candidates, number of ballots, calculations, how many votes a candidate had, winner(s), and election progression. It will also show who got what ballot and the ballots order of being received if applicable.

The following is a draft of the audit file for IR voting:

```
Run Date: xxx
Election Type: IR
# of candidates: xxx
Candidates: xxx
# of initial ballots: xxx
```

```
Winner(s): xxx
```

```
-----
Round 1 of 2
```

Candidate	CandidateA	CandidateB	CandidateC
Votes	1234	432	5147
Candidate removed: CandidateB			
Tie Broken: No			

```
Round 2 of 2
```

Candidate	CandidateA	CandidateC
Votes	1434	5347
Candidate removed: CandidateA		
Tie broken: No		

```
Ballots Received By Each Candidate
```

```
Candidate A
1,2,5,7,12,...
Candidate B
3,13,21,22,...
Candidate C
4,8,11,19,...
```

The following is a draft of the audit file for CPL voting:

Run Date: xxx

Election Type: CPL

of parties: xxx

Candidates: xxx

of initial ballots: xxx

of seats: 2

Winner(s): xxx

Round Whole Seat Assignment

Party	PartyA	PartyB	PartyC
Votes	1234	432	5147
Seats Voted	0.36	0.13	1.51
Seats Assing	0	0	1
Remainder	0.36	0.13	0.51

Round Remainder Seat Assignment

Party	PartyA	PartyB	PartyC
Remainder	0.36	0.13	0.51
Seats Assing	0	0	1
Tie broken: No			
Total Seats	0	0	2

Ballots Received By Each Party

Party A

1,2,5,7,12,...

Party B

3,13,21,22,...

Party C

4,8,11,19,...

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system must finish the process for 100,000 ballots within 4 minutes (240 seconds), excluding the time elapsed while waiting for user input.

5.2 Safety Requirements

There are no explicit safety requirements for this system.

5.3 Security Requirements

- The input ballot file cannot be modified outside of the program.
- Election officials are allowed to gather data from the result file in order to finalize the election.
- When creating an audit file for any election, the system must fully secure the file by disabling two permissions: writing and executing.
- Modifications to both files are completely prohibited.
- The system must not contain any hidden implementation that allows internal adjustments to either one of them.
- All the data read from the ballot file must match the data written into the audit file.

5.4 Software Quality Attributes

- There must not be any error in the tabulation of votes, or in the results derived from that tabulation.
- The system must be readily usable by election officials, some of whom may not be particularly experienced with computers or modern software applications.
- The system must have minimum user input to ensure correctness, accuracy, and legitimacy of results tabulated.
- The system must have auditable results and any fair random calculations must take precautions to avoid computer random biases as much as possible.
- There will be no numbering mistakes in the input files.

5.5 Business Rules

There are no business rules which explicitly define product requirements. All users will be granted the same access to software functionality.

6. Other Requirements

No additional requirements are explicitly specified for this system.

Appendix A: Glossary

- IR: Instant Runoff voting method
- CPL: Closed Party List voting method
- Election officials: the authorized people in charge of initiating the system and finalizing the results
- Tie-breaker: method(s) to determine the winner between two or more candidates who have the same amount of votes

Appendix B: Analysis Models

There are no analysis models for this system.

Appendix C: To Be Determined List

There is no TBD list for this system.