
Software Requirements Specification

for

IRV and OPL Voting System

Version 1.0 approved

Max Bryson, bryso049, Team12

Hannah Nelson, nel01283, Team12

Ruichen He, he000239, Team12

Christopher Miao, miao0079, Team12

CSCI 5801, University of Minnesota

September 2023

Table of Contents

Table of Contents	ii
Revision History	ii
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	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
4. System Features	4
4.1 System Feature 1	4
4.2 System Feature 2 (and so on)	4
5. Other Nonfunctional Requirements	4
5.1 Performance Requirements	4
5.2 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	5
5.5 Business Rules	5
6. Other Requirements	5
Appendix A: Glossary	5
Appendix B: Analysis Models	5
Appendix C: To Be Determined List	6

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to provide a description of the vote counting system, which can use instant runoff voting and party list voting. The system will be integrated with a pre-existing ballot system and is only responsible for reading ballots from instant runoff voting and open party list elections. This document will explain features and interfaces, as well as the expected output, and the constraints under which it must operate.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents. The abbreviation IRV is used for Instant Runoff voting and OPL for Open Party List when referring to ballot/election types.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- Typical users such as election officials who want to use the system to quickly count votes and determine a winner of an IRV or OPL election. These readers will be most interested in sections 2.2 for a description of system functions, 2.6 for a list of other user documentation, and 3.1 for a description of the user interface.
- Programmers who wish to further develop the system or fix existing bugs. These readers will want to read the entire document.
- Potential media personnel who want to publicize the result of an election and its process.

1.4 Product Scope

This vote counting system is designed to read a formatted .csv containing voting information for IRV or OPL voting. The objective of this software is that users will be able to enter the name of the file contained in the same directory of the program and the system will process the information on the file to determine a winner. Benefits of this software include: determination of winner and voter breakdown.

1.5 References

IEEE template for System Requirement Specification Documents:

<https://goo.gl/nsUFwy>

GitHub Page:

<https://github.umn.edu/umn-csci-5801-03-F23/repo-Team12>

2. Overall Description

2.1 Product Perspective

This voting system was developed for election officials to quickly and easily determine winners from IRV and OPL elections. Results of elections are recorded into a CSV file that this system is capable of processing. Generating and formatting these files is outside the scope of the system and is handled by a different program.

2.2 Product Functions

- Collect user input from terminal for the name of ballot file
- Open and read CSV
- Determine if CSV is an IRV or OPL ballot
- Calculate a winner
- Output the winner to terminal
- Output an audit file

2.3 User Classes and Characteristics

The primary user of this system is an election official. The official will only be able to enter the full name of the csv file (with .csv extension) and then get a winner. If the user enters the filename incorrectly the system will re prompt the user. Secondary users would be developers looking to add functionality or eliminate bugs.

2.4 Operating Environment

Software must operate on Ubuntu 20.04 on a CSE labs machine. The software may run on other systems but will not be tested on them.

2.5 Design and Implementation Constraints

The system is developed with C++. The system must be able to process 100,000 ballots in under 8 minutes.

2.6 Assumptions and Dependencies

The voting system is developed with C++ and will require the users to have C++ is installed on their system. This system has been developed on the latest version of C++ as of 9/27/2023.

3. External Interface Requirements

3.1 User Interfaces

User Interface will be through the terminal and text prompts. Users will need to input the complete filename of the ballot including extension Ex: ballot1.csv

3.2 Hardware Interfaces

Software must run on a CSE labs machine. While it may run on other systems, the CSE lab machine will be the only one used for testing.

3.3 Software Interfaces

The system is written in C++ and will require C++ to be installed on the machine. Additional information can be found in section 2.6.

4. System Features / Use Cases

Name	Read User Filename
ID	UC_001
Description	Read in and collect user input for a filename
Actors	Election Official
Organizational Benefits	Gives user control over which election to count
Frequency of Use	Every time an instant runoff or open party listing election is held and election officials want to determine the result
Triggers	User starts program
Preconditions	None
Postconditions	Filename is read in
Main Course	<ol style="list-style-type: none"> 1. System prompts user for filename 2. File is found (see AC1) 3. Program continues
Alternative Courses	AC1 User enters invalid filename <ol style="list-style-type: none"> 1. Return to Main Course step 1
Exceptions	None

Name	Open File
ID	UC_002
Description	Open vote file for reading
Actors	Election Official
Organizational Benefits	Opens one file at a time so results aren't mixed
Frequency of Use	Every IRV or OPL election
Triggers	Filename is inputted in UC_001
Preconditions	Filename is valid and the file exists
Postconditions	File is open for reading and processing
Main Course	1. Open file for reading
Alternative Courses	AC1 file is not a comma separated values file 1. Return user to UC_001
Exceptions	None

Name	Identify Voting Type
ID	UC_003
Description	Based on first line of the file, determines if the election is IRV or OPL voting
Actors	Computer program
Organizational Benefits	Allows for program to branch easily for 2 different voting formats within the same program
Frequency of Use	Every IRV or OPL election
Triggers	File is open for reading and first line is read
Preconditions	File must be open, readable and is a comma separated values file
Postconditions	Program knows which voting type to use
Main Course	1. First line of file is read 2. System determines if it is IRV or OPL 3. Instant runoff voting or open party list voting is selected
Alternative Courses	None

Exceptions	EX1 First line of the file is not IRV or OPL. 1. Return user to UC__001
-------------------	--

Name	Create Audit File
ID	UC_004
Description	Create Audit File for recording election information
Actors	Computer Program
Organizational Benefits	Records and shows how the election was calculated for verification and documentation
Frequency of Use	Every instant runoff and open party list election
Triggers	Election type is selected
Preconditions	File has been opened and the election type has been read in
Postconditions	Audit File is created and open for writing
Main Course	1. Audit File is created 2. Audit File is opened for writing
Alternative Courses	None
Exceptions	EX1 The CSV file has no data or incorrect data format after the first line 1. Return user to UC_001

Name	Display Stats / Calculations
ID	UC_005
Description	Display steps in calculating vote
Actors	Election Official
Organizational Benefits	Allows users to see what the program is doing and verify it
Frequency of Use	Every IRV or OPL vote
Triggers	During calculation of IRV or OPL vote
Preconditions	Ballot has been read in and Identified
Postconditions	Output to screen
Main Course	1. File is read in

	<ol style="list-style-type: none"> 2. Calculations are done 3. Information is printed to screen
Alternative Courses	None
Exceptions	None

Name	Coin Toss Is Fair
ID	UC_006
Description	If there is a tie amongst the winners, a fair coin will be tossed. Each winner has the same chance to win in this tie-breaker.
Actors	Election Official
Organizational Benefits	Produces a winner when there would have been none due to the winners getting the same amount of votes.
Frequency of Use	Some IRV or OPL election
Triggers	A tie occurs at the end of counting votes.
Preconditions	Final candidates have equal votes
Postconditions	A winner is decided.
Main Course	<ol style="list-style-type: none"> 1. There is a tie between final candidates 2. A coin is tossed 3. A winner is declared
Alternative Courses	None
Exceptions	None

Name	IRV Popularity Wins If No Clear Majority
ID	UC_007
Description	In an instant runoff election, if there is no majority, then the most popular candidate wins the election.
Actors	Election Official
Organizational Benefits	Allows a winner to be decided when there is no candidate who has a majority vote.
Frequency of Use	Some IRV elections.

Triggers	No majority at the conclusion of an IRV vote
Preconditions	At the end of an IRV voting count, no candidate has a majority vote
Postconditions	A winner is declared
Main Course	<ol style="list-style-type: none"> 1. IRV voting concludes with no majority vote winner 2. The vote percent is calculated for each candidate 3. The candidate with the higher percent is declared (EX1)
Alternative Courses	None
Exceptions	EX1 - Tie between candidates: <ol style="list-style-type: none"> 1. See UC_006

Name	Calculate the vote count for IRV in specific round, and declare the winner if possible
ID	UC_008
Description	Calculate the current vote count for all remaining candidates
Actors	Election Official
Organizational Benefits	Get the count for all candidates in order to decide a winner
Frequency of Use	In the beginning of the vote counting steps, and in each round when there is no candidates have received more than 50% votes
Triggers	If the voting type is IR
Preconditions	Voting type is IR
Postconditions	A winner is declared, or a new round of voting count begin with the candidates with fewest support removed from the pool
Main Course	<ol style="list-style-type: none"> 1. The voting type is IR 2. The count of the vote for each participant is calculated 3. A candidate has more than 50% voters 4. A winner is declared
Alternative Courses	AC1: <ol style="list-style-type: none"> 1. The voting type is IR 2. The count of the vote for each participant is calculated 3. No candidate has more than 50% voters, and there is only one candidate with fewest voters 4. Remove the candidate with fewest supporter, transfer the vote (UC_009), and recalculate the vote AC2: <ol style="list-style-type: none"> 1. The voting type is IR 2. The count of the vote for each participant is calculated 3. No candidate has more than 50% voters, and there is no candidate with fewest voters

	4. Decide the winner based on popularity (UC_007)
Exceptions	None

Name	Remove the candidate with fewest supporter in IR, and translate the vote
ID	UC_009
Description	When there is no candidate with 50% more voters support, remove the candidate with fewest voters from the pool, and transfer the vote from the voters of the removed candidate
Actors	Election Official
Organizational Benefits	Since no candidate win in the current round, removing the candidate with fewest voters and recounting the vote is the necessary step to get a winner
Frequency of Use	For some IR, especially in the initial counting steps
Triggers	If current counting round has no winner with more than 50% voters
Preconditions	If the voting type is IR, and in the current counting step, no candidate get more than 50% voters
Postconditions	A new candidate pool is formed, and is ready for counting
Main Course	<ol style="list-style-type: none"> 1. From the counting, the count for each candidate is known 2. Based on the counting, the candidate with fewest voters is removed from the pool 3. The votes of those voters who support the removed candidates are adjusted
Alternative Courses	None
Exceptions	If there is no other candidate that the voter supports, then ignore those voters in the following counting steps

Name	Calculate the Quota for OPL
ID	UC_010
Description	If the vote type is OPL, based on the total number of seats, and total number of votes, calculate the quota that will be used in the largest remainder formula.
Actors	Election Official
Organizational Benefits	Will be useful when using largest remainder formula to calculate the winner
Frequency of Use	For each OPL election

Triggers	If the vote type is OPL
Preconditions	Vote type is OPL
Postconditions	A quota is calculated, and system is ready to calculate the real winner
Main Course	<ol style="list-style-type: none"> 1. The vote type is OPL 2. Calculate the total vote 3. Based on the input total number of seats and the total vote, calculate the quota
Alternative Courses	None
Exceptions	None

Name	Calculate the Winner of OPL
ID	UC_011
Description	If the vote type is OPL, calculate the winner using largest remainder formula
Actors	Election Official
Organizational Benefits	Will be useful when using largest remainder formula to calculate the winner
Frequency of Use	For each OPL election
Triggers	If the vote type is OPL
Preconditions	Vote type is OPL, and quota is calculated
Postconditions	Winners of the vote are determined
Main Course	<ol style="list-style-type: none"> 1. Based on the calculated quota, and the count of the vote, decide the first allocation of seats 2. Based on the remaining votes, decide the second allocation of seats 3. Calculate the final seats allocation, and decide the candidate among each party that get the seats
Alternative Courses	None
Exceptions	None

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The election system must swiftly process large data volumes, being capable of handling 100,000 ballots in under 8 minutes. This requirement ensures timely results, maintaining public trust and promoting transparency.

5.2 Safety Requirements

One primary safety requirement is ensuring the system can detect any anomalies or corruptions in the file format that might arise from accidental damage or potential tampering. If detected, the system should halt processing and alert the relevant authorities or administrators.

5.3 Security Requirements

While primary security measures like one-person-one-vote authentication are managed at the voting centers, it's imperative for the election system to maintain the integrity of data. Specifically, once a file is introduced to the system for processing, it must be safeguarded against any unauthorized modifications.

5.4 Software Quality Attributes

The system should be robust, gracefully alerting to anomalies without crashing, and maintain a user-friendly interface ensuring the majority of users can process files with minimal attempts.

Appendix A: Glossary

IRV = Instant Runoff, a type of voting system used for elections

OPL = Open Party List, a type of voting system used for elections.