Software Requirements Specification

for

Election Voting System

Version 1.0 approved

Evan Bagwell, Andrew Guerra, Hady Kotifani, Tyler Grimm

CSCI 5801 - Software Engineering I

February 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Initial Version	2/15		0.1

1. Introduction

1.1 Purpose

The purpose of this document is to present an in-depth description of the voting system software. It will explain the purpose and features of the software, what the voting system software will do including how it interacts under various use cases, and the variety of constraints under which it must function.

1.2 Document Conventions

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

1.3 Intended Audience and Reading Suggestions

- Election Officials and Auditors–those who intend to run elections or audit elections–should reference
    - Section(s) 2.1, 3.1
- Tester– those who intend to test functionality and components–should reference
    - All sections of this document
- Programmers–those who intend to implement the system or fix existing bugs–should reference
    - All sections of this document

1.4 Product Scope

The Voting System Software is designed to determine election results for either Instant Runoff (IR) or Closed Party List (CPL) elections. This system will be designed to be fair in its election determinations. The benefits of the Voting System Software is that it allows multiple election types to be processed through a single program and also removes manual processes by automating the process of counting votes and determining winners. The system is designed to be simple with guided prompts to aid Election Officials in running the process.
Additionally, the system will produce an audit file containing information about winners and the processing of ballots so that media and auditors may review how the election unfolded. It's purpose is to instill trust in the system through transparency and legitamize the results of the election. The long-term goal of the Voting System Software is to eventually be integrated into an online voting system.

1.5 References

Voting Type References:

https://fairvote.org/

IEEE Template for System Requirements Specification Document

https://goo.gl/nsUFwy

2. Overall Description

2.1 Product Perspective

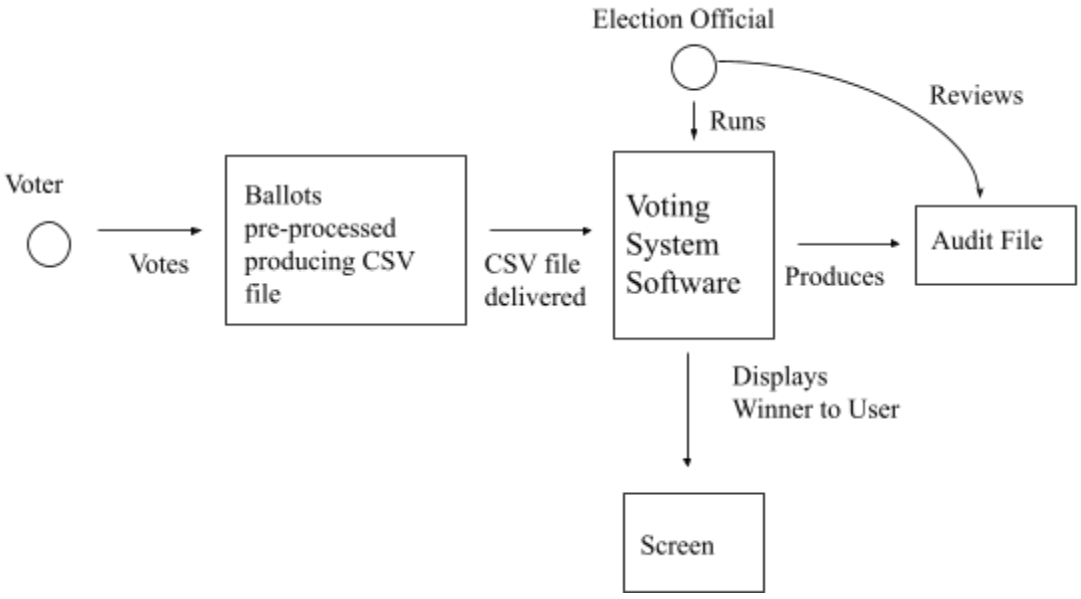


Figure 1: System Interactions

The Voting System Software is a self-contained system that will determine the winners involved in either Instant Runoff Voting or Party List Voting using a Closed Party List System. The product is developed for Election Officials to automate the process of determining the winners of an election after the ballots are accounted for. The actual voting process is done separately and ballots are preprocessed before the Voting System Software runs. The Voting System Software reads in the pre-processed comma-delimited (CSV) file that contains information about the casted ballots; then, from the file, the Voting System Software determines the type of election specified and determinse the corresponding winner(s). A summary of the election that includes the winner(s), the type of election, and the number of seats will be displayed on the screen for the user. The System shall also produce an audit file with election information that contains the winners, a detailed history showing who got what ballot (and order of ballot's being received if applicable), and details

about the election (e.g. Type of Voting, Number of Candidates, Candidates, Number of Ballots, calculations, how many votes a candidate had, etc) upon completion of the election.

2.2 Product Functions

- Accept a file: users will input a file containing all information needed for the election, the system will check for correct inputs.
- Read the file: the system will read the file and store the information contained in the file.
- Determine Election Type: the system will determined the type of election based on the contents of the previously read file
- Run election: the system will run the previously determined election.
- Tally Ballots: the system will tally all of the ballots in the file
- Calculate largest remainder: the system will calculate the largest remainder for use in CPL voting.
- Determine loser: the system will determine the loser in each round of IR voting
- Redistribute ballots: the system will distributed the losers ballots in each round of IR
- Fair Coin Toss: the system will flip a coin to determine a winner if there is a tie.
- Determine a Winner: the system will determine a winner based on the results of the tally and the coin toss if needed.
- Display Summary: the system will display a small summary of the election results.
- Generate Audit File: the system will generate an in depth report about the election.

2.3 User Classes and Characteristics

- **Election Officials**, most typical of Users, will use the Voting System Software to analyze election results and determine the winner of either IR or CPL elections. Election Officials should have basic knowledge of computers and the general use of a computer (eg. using a keyboard and mouse). Users of this system are also expected to understand how to run the program and respond to prompts. Election Officials should understand how the produced Audit file is formatted (the information and file formatting, being a .txt) and understand what each part of the audit file means so that information about the election can be analyzed and shared.
- **Programmers and Testers** who use this product should be familiar with java programming and terminal commands in order to fix bugs, add additional features, or run the program. Additionally, programmers and testers should be familiar with the process of IR and CPL elections so that the algorithms and code produced follow the expected results for the corresponding election type.

2.4 Operating Environment

The system will be run on a computer terminal that is using a Linux operating system (CSE Lab Machine).

- Linux OS
- Versions?
- Eclipse or other Java IDE?
- If you provide a Java program, you must provide all source files and class files. Java programs will be run either at the command prompt or through Eclipse.

2.5 Design and Implementation Constraints

The system is designed for Linux systems and is not expected to behave or run properly under other operating systems. The system is developed in Java and is not expected to interact with other languages. The system must be able to run

Since the system was built and tested on a computer that is running Linux there is no guarantee that the system will function properly on an operating system other than Linux. The system is developed in Java and all source and class files will be provided.

2.6 User Documentation

This document will be delivered with the software and users will be able to refer to this document with any questions about the system. The system will also feature a help window that the user can refer to to help understand how to use the system.

2.7 Assumptions and Dependencies

It is assumed that:

- All ballots will be a comma separated values (.csv) file where each row is separated by a newline.
- The file will be exported from Excel into the .csv format and all preprocessing of the file will be done prior to being placed in the directory.
- The first line of the file will list the type of voting that the system needs to run (IR, CPL).
- No more than one file per election will be given to the system.
- For CPL, all independent groups will only have a single member.
- For IRV, the ballot will have at least 1 person ranked.
- For IRV, a candidate will be given only 1 ranking.
- Security such as ensuring one vote for one person is handled at the voting centers.

- You can assume we will use the most up-to-date CSELabs machines.
- There will never be more than one file given to you per election.
- No write ins
- How IR works -> Each ballot guaranteed to have 1 person ranked

- You may assume that there are no numbering mistakes in the file (e.g. a voter will not make any mistakes on the ballot.)

3. External Interface Requirements

3.1 User Interfaces

The voting system will have a user interface in the terminal that prompts the user with various courses of action.

3.2 Hardware Interfaces

The voting system will not have any external hardware interfaces to worry about. Only a computer is needed.

3.3 Software Interfaces

The voting system will need to be able to accept and parse an excel (.csv) file that will be read in from the same working directory the system is in.

3.4 Communications Interfaces

The voting system will not need to communicate with any outside resources. All ballot files (.csv) will be in the current working directory. The system will not need an internet connection in order to run correctly.

4. System Features

4.1 Read in Election File

Name	Read in Election File
ID	UC_4.1
Description	A file will be inputted from the command line when the program is run or the program will prompt the user in the terminal for a file name

Actors	Election Official, Programmer, Auditor, Tester
Organizational Benefits	Eases the ways in which Election Officials and Programmers can read in a filename where Election Officials do not need to learn command line operations while Programmers can fast track inputting file names within the command line.
Frequency of Use	Once per election
Triggers	The user has run the program
Preconditions	The election file to be inputted exists in the same directory The election file is a well formatted CSV file There is only one election file to be processed Election file cannot be modified prior to reading contents
Postconditions	The election type, candidates (and parties), number of ballots, and ballots have been read and determined from the provided file name or the terminal
Main Course	1. Filename will be read as argument from command line 2. Text file will be read in, program will acknowledge success of the read in file 3. Failure of file read in will result in (EX1) 4. Determine if election parameters will be supplied from the supplied file or through the terminal (UC_4.2) 5. Determine election type (UC_4.3) 6. Determine candidates if IR election (UC_4.4) or Determine party and candidates if CPL election (UC_4.5) 7. Determine number of ballots (UC_4.6) 8. Read IR ballots if IR election (UC_4.7) or Read CPL ballots if CPL election (UC_4.8)
Alternate Course	AC1 User Runs Program with No File Input 1. Program prompts user for filename 2. User inputs filename 3. Text file will be read in, program will acknowledge success of the read in file
Exceptions	EX1 Invalid Input from command line or prompt 1. Display error to user that invalid file name was provided 2. Return user to MC

4.2 Determine Automatic or Manual Election Parameter Input

Name	Automatic or Manual Election Parameter Selection
ID	UC_4.2
Description	User will be prompted with the choice of reading in election parameters from the file or manually via a prompt in the terminal
Actors	Election Official, Programmer, Auditor, Tester
Organizational Benefits	Determining manual or automatic read in of election parameters can help speed up election processing and allows for greater user flexibility with parameters.
Frequency of Use	Once per election
Triggers	The user has run the program and has successfully read in an election file
Preconditions	The file to be inputted exists in the same directory The file is a CSV file
Postconditions	The user will be prompted with the selection made
Main Course	1. User will be prompted with the choice of reading in election parameters manually or automatically 2. User inputs selection of manual or automatic input of election parameters 3. Failure to read valid input results in (EX1) 4. User is prompted with confirmation of selection
Alternate Course	N/A
Exceptions	EX1 Invalid Input from command line or prompt 3. User is prompted with error message 4. Return user to MC

4.3 Determine Election Type

Name	Determine Election Type

ID	UC_4.3
Description	The system will determine the correct voting algorithm that needs to run based off the value in the file
Actors	Election Official, Tester, Auditor, Programmer
Organizational Benefits	This will allow the system to determine what algorithm should be run so it can tally votes in the correct manner.
Frequency of Use	Once per election
Triggers	The system will decide what type of voting to run based on what is contained in the text file or given by the user.
Preconditions	A valid file has been read into the system.
Postconditions	A unique value that correspond with either the CPL or the IR will be stored in the system and will be used run the correct algorithm
Main Course	1. File reads first line from CSV file 2. Check if either IR or CPL election 3. Program displays election type from file
Alternate Course	AC1: Manual Input 1. User gives election type through prompt 2. Program displays election type from file
Exceptions	EX1: Invalid Manual Election Type Input 1. Display error message if user enters value that does not correspond to either IRV or CPL 2. Return to (AC1)

4.4 Determine Candidates IR

Name	Determine Candidates for IR Election
ID	UC_4.2.2
Description	Reads in or prompts the user for the candidates in IR that are being voted for
Actors	Election Official, Tester, Auditor, Programmer

Organizational Benefits	Allows the system to determine the number and names of the candidates present in an IR election
Frequency of Use	Once per election
Triggers	IR election has been determined from the user or file
Preconditions	Determined that the election is IR
Postconditions	The name of all candidates are stored in the system
Main Course	1. Reads in the second line of the file for number of candidates 2. Reads in the third line of the file to determine candidate names given how many candidates 3. Program displays candidate names to the user
Alternate Course	AC1: Manual Input 1. User manually inputs number of candidates that are present for the IR election 2. User manually inputs the name of the candidates in the order they appear on the ballot
Exceptions	EX1: Invalid Input 1. Display error message if user enters value or names that do not correspond with the IR election 2. Return to (AC1)

4.5 Determine Parties and Candidates CPL

Name	Determine Parties for CPL
ID	UC_4.2.3
Description	Reads in or prompts the user for the parties in CPL that are being voted for. Independent candidates will belong to groups consisting of only that candidate.
Actors	Election Official, Tester, Auditor, Programmer
Organizational Benefits	Allows the system to determine the number and names of parties present in CPL and correctly map votes from ballots to each party. Ensures the count and corresponding party name that gets seats are accounted for.
Frequency of Use	Once per election

Triggers	CPL election has been determined from the user or file
Preconditions	Determined that the election is CPL
Postconditions	The name of all parties are stored in the system and ordered correctly
Main Course	1. Reads in the second line of the file for number of parties 2. Reads in the third line of the file to determine party names given number of parties 3. Read in names of candidates for their respective party and store
Alternate Course	AC1: Manual Input 1. User manually inputs number of parties that are present for the CPL election 2. User manually inputs the name of the parties in the order they appear on the ballot
Exceptions	EX1: Invalid Input 1. Display error message if user enters value or party names that do not correspond with the CPL election 2. Return to (AC1)

4.6 Determine Number of Ballots

Name	Determine Number of Ballots
ID	UC_4.2.4
Description	Reads in or prompts the user for the number of ballots in the election file
Actors	Election Official, Tester, Auditor, Programmer
Organizational Benefits	Allows Election Officials to see if the number of Ballots corresponds to what is expected by the election. Allows the program to read in the corresponding number of ballots
Frequency of Use	Once per Election
Triggers	Determine candidates/parties in the election
Preconditions	Determined election type Determined candidates or parties
Postconditions	Number of ballots are kept/known

Main Course	1. Reads in the number of ballots from file 2. Program displays ballot count to the user
Alternate Course	AC1: Manual Input 1. User manually inputs the number of ballots to be read from the election file
Exceptions	EX1: Invalid Input 1. Display error message if user enters invalid input 2. Return to (AC1)

4.7 Read IR Ballots

Name	Read IR Ballots
ID	UC_4.3
Description	Read the ballots from the election file. Ballots will be provided in a csv format after the header specified in (see specification), with each line representing one ballot. Each value will represent the ranking, with a value of 1 to the number of candidates. The position will indicate the candidate associated with each ranking. All rankings on each ballot will be unique, but not all rankings from 1 to the number of candidates need to be present for each ballot. All ballots will have at least one candidate ranked. No write-in candidates are allowed.
Actors	Election Official, Tester, Auditor, Programmer
Organizational Benefits	The order in which ballots are tallied has an effect on determining the winner. Shuffling ballots ensures that the process is treated fairly and increases the validity of the election.
Frequency of Use	Once per Election
Triggers	Number of ballots determined
Preconditions	IR election type confirmed, candidates/parties are known, ballots follow the format specified
Postconditions	Information from all ballots are stored
Main Course	Read in the ballots from the election file and store it in the system

Alternate Course	N/A
Exceptions	N/A

4.8 Read CPL Ballots

Name	Read CPL Ballots
ID	UC_4.4
Description	Read the ballots from the election file. Ballots will be provided in a csv format after the header specified in (see specification), with each line representing one ballot. The position of a value will indicate the preference for a party at the corresponding position. All values will be 1.
Actors	Election Official, Tester, Auditor, Programmer
Organizational Benefits	The order in which ballots are tallied has an effect on determining the winner. Shuffling ballots ensures that the process is treated fairly and increases the validity of the election.
Frequency of Use	Once per Election
Triggers	Number of ballots determined
Preconditions	CPL election type confirmed, candidates/parties are known, ballots follow the format specified
Postconditions	Information from all ballots are stored
Main Course	Read in the ballots from the election file and store it in the system
Alternate Course	N/A
Exceptions	N/A

4.9 Shuffle Ballots

Name	Shuffle Ballots
ID	UC_4.4
Description	Shuffle the order of the ballots
Actors	Election Official, Tester, Auditor, Programmer
Organizational Benefits	The order in which ballots are tallied has an effect on determining the winner. Shuffling ballots ensures that the process is treated fairly and increases the validity of the election.
Frequency of Use	Once per Election
Triggers	Ballots have been read into the system
Preconditions	Election type is known candidates/parties are known Ballots count is known Ballots have been read into the system
Postconditions	Information from all ballots are stored and the order in which they are tallied has been randomly determined.
Main Course	Shuffle the order of the ballots in the system
Alternate Course	N/A
Exceptions	N/A

4.10 Run IR Election

Name	Run IR Election
ID	UC_4.4
Description	The system will run the election using the rules involved in an IR election
Actors	Election Official, Programmer, Auditor, Tester
Organizational Benefits	Runs the series of steps to tally up votes in an IR election

Frequency of Use	Will run once per IR election
Triggers	Ballots have been read into the system and shuffled and the election type is an IR election
Preconditions	Election type is IR All ballots have been read into the system Ballots have been shuffled
Postconditions	IR election winner has been determined
Main Course	1. Tally votes according to IR rules (UC_4.12) 2. Declare election winner that candidate has a majority of the votes after considering all first ranked votes
Alternate Course	AC: Election has not been won 1. Eliminate candidate with the fewest votes (UC_4.5.2) 2. Retally votes, go to main course step one
Exceptions	N/A

4.5.1 Check for IR Election Win

Name	Check for IR Election Win
ID	UC_4.5.1
Description	Check if the IR Election has been won
Actors	Election Official, Programmer, Auditor, Tester
Organizational Benefits	N/A
Frequency of Use	Multiple times in IR election
Triggers	An IR election is being run
Preconditions	In an IR election Votes are Tallied
Postconditions	Whether a candidate has won has been verified
Main Course	1. Check each candidate for a majority 2. Declare election winner (UC_4.7)

Alternate Course	AC: Two Candidates left in election with neither having a majority of votes 1. Check for candidate with plurality 2. Declare election winner (UC_4.7) AC: Two Candidates left in election with same amount of votes 1. Fair coin toss elimination (UC_4.8) 2. Declare election winner (UC_4.7) AC: More than two candidates and no majority win 1. Eliminate candidate with the least number of ballots (UC_4.5.2) 2. Retally votes (see UC_4.5 for next steps)
Exceptions	N/A

4.5.2 Eliminate Candidate

Name	Eliminate Candidate with Fewest Ballots
ID	UC_4.5.2
Description	In Instant runoff voting, the candidate with the fewest votes is removed
Actors	Election Official, Programmer, Auditor, Tester
Organizational Benefits	N/A
Frequency of Use	Occurs whenever there lacks a clear majority in IR voting
Triggers	The IR algorithm lacks a clear majority
Preconditions	There is no clear majority in IR Number of votes each candidate has is known
Postconditions	Candidate has been removed
Main Course	1. Find Candidate with least number of votes 2. Candidate with the least number of votes is removed 3. Retally votes, go to (UC_4.5) step one
Alternate Course	AC1: Multiple Candidates with Fewest Number of Votes 1. Eliminate one candidate determined by a fair coin toss (UC_4.8) 2. Retally votes, go to (UC_4.5) step one

Exceptions	N/A

4.11 Run CPL Election

Name	Run CPL Election
ID	UC_4.6
Description	The system will run the election using the rules involved in an CPL election Seats are distributed to party in specific order.
Actors	Election Official, Programmer, Auditor, Tester
Organizational Benefits	Runs the series of steps to tally up votes in an CPL
Frequency of Use	Once per CPL Election
Triggers	Ballots have been read into the system and the election type is a CPL election
Preconditions	Election type is CPL All ballots have been read into the system Ballots have been shuffled
Postconditions	Seat allocation has been determined
Main Course	1. Read in ballots stored from the system from (UC_4.3) 2. Tally votes according to IR rules (UC_4.7) 3. Determine quota (UC_4.6.1) 4. First round allocation of seats (UC_4.6.2) 5. Remainder allocation of seats (UC_4.6.3) 6. Display election results (UC_4.7)
Alternate Course	AC1: Seats Leftover after Remainder Allocation of Seats 1. Leftover seat delegation (UC_4.6.4) 2. Display election results (UC_4.7)
Exceptions	N/A

4.6.1 Calculate Quota

Name	Calculate Quota
ID	UC_4.6.1
Description	Determines the number of votes required to win a seat
Actors	Election Official, Programmer, Auditor, Tester
Organizational Benefits	Ensures that each party gets a seat that is roughly proportional to the number of ballots in the district
Frequency of Use	Once per CPL election
Triggers	CPL election is ran
Preconditions	The total number of seats available are known The total number of Ballots are known
Postconditions	The quota (number of votes required) to win a seat is determined
Main Course	MC1: 1. Divide the total number of ballots by the total number of seats to determine quota (EX1)
Alternate Course	N/A
Exceptions	EX1: Quota is not a Whole Number 1. Round quota value to the next highest whole number to prevent seat underflow

4.6.2 First Round Allocation of Seats

Name	First Round Allocation of Seat
ID	UC_4.6.2
Description	Party is allocated seat(s) determined from the number of votes that meet the quota
Actors	Election Official, Programmer, Auditor, Tester
Organizational Benefits	The election initially allocates seats in a way that is proportionally representative of the district
Frequency of Use	Once per CPL election

Triggers	CPL election is ran
Preconditions	Number of votes per party is known Quota is known
Postconditions	Initial number of seat allocations per party Remainder of votes for each party is known Number of seats remaining is known
Main Course	MC1: 1. Divide the number of votes for each party by the quota to determine seats allocate 2. For each whole number, allocate one seat to the party 3. Calculate the remainder of votes for each party based on quota 4. Keep track of remaining seats
Alternate Course	N/A
Exceptions	N/A

4.6.3 Remainder Allocation of Seats

Name	Remainder Allocation of Seats
ID	UC_4.6.3
Description	Allocates seats to the parties with the highest remaining votes
Actors	Election Official, Programmer, Auditor, Tester
Organizational Benefits	akljfhaklsdhf
Frequency of Use	Once per CPL election
Triggers	CPL election is ran
Preconditions	Number of seats remaining known Number of remainder votes for each party is known
Postconditions	All the seats have bee
Main Course	MC1: 1. Rank the parties from highest to lowest based on their respective number of remaining votes

	2. Delegate remaining seats in order of parties with the highest votes to lowest votes (see AC1 if tie and AC2 if party does not have candidate available for winning seat) 3. Allocate any leftover seats (UC_4.6.4)
Alternate Course	AC1: Remaining Votes between Two or More Parties the Same 1. Delegate remaining seats to party determined by fair coin toss (UC_4.8) AC2: Remainder Seat(s) Delegated to Party Does Not Have a Candidate 1. Leftover Seat Delegation (UC_4.6.4)
Exceptions	N/A

4.6.4 Leftover Seat Delegation

Name	Leftover Seat Delegation
ID	UC_4.6.4
Description	The system shall delegate any remaining seats to parties by fair coin toss
Actors	Election Official, Programmer, Auditor, Tester
Organizational Benefits	Ensures that all seats are delegated
Frequency of Use	Once per CPL election
Triggers	CPL election is run
Preconditions	Number of leftover seats known Parties that have available candidates known
Postconditions	All seat delegations have been determined
Main Course	MC1: 1. Conduct a fair coin toss on remaining parties to determine ordering of seats to be delegated for parties that have candidates remaining (UC_4.8) 2. Delegate seats to parties based on ordering (AC1)
Alternate Course	AC1: Party(s) do not have enough candidates for seat allocation after coin toss 1. Delegate the seats available to parties based on relative ordering 2. Conduct another fair coin toss to delegate remaining seat(s) (go back to MC1)

Exceptions	N/A

4.12 Tally Votes

ID	UC_4.7
Description	Determines the number of votes each candidate/party has received
Actors	Election Official, Programmer, Auditor, Tester
Organizational Benefits	
Frequency of Use	Multiple times an election cycle
Triggers	An election is run
Preconditions	Election type has been determined Ballots have been read in and stored in the system
Postconditions	Number of votes per candidate or party is known
Main Course	MC: IR Election Tally 1. Count the number of votes each candidate received MC: CPL Election Tally 1. Count the number of votes each party received
Alternate Course	N/A
Exceptions	N/A

4.13 Fair Coin Toss Winner

Name	Fair Coin Toss Winner
ID	UC_4.8
Description	Determines winner from a fair coin toss
Actors	Election Official, Programmer, Auditor, Tester
Organizational	Ensures that the election process is fair and instills trust and integrity in

Benefits	the system.
Frequency of Use	Multiple times per election
Triggers	A tie has been determined
Preconditions	A tie has occurred between votes or seats for a candidate/party Known how many
Postconditions	Seats have been marked as allocated to winning parties
Main Course	MC1: IR Two Candidate Plurarity Tie 1. Assign random number to each candidate 2. Generate a random number that corresponds to a candidate to simulate a fair a coin toss) 3. Determine winner from "coin toss" (UC_4.9) MC2: IR Candidate with Fewest Votes Tie 1. Assign random number to each candidate 2. Generate a random number that corresponds to a candidate to simulate a fair a coin toss) 3. Determine loser from "coin toss" and eliminate candidate (see UC_4.5.2 and UC_4.5) MC3: CPL -> Remainder votes ties (Make these alts?) MC4: More Seats than allocated
Alternate Course	N/A
Exceptions	None

4.14 Display Election Results to Terminal

Name	Display Winner of Election
ID	UC_4.9
Description	Display the winner of the election after a winner has been determined
Actors	Elections Official
Organizational Benefits	Allows for easy viewing of a winner after an election has been run
Frequency of Use	Once

Triggers	The election winner(s) have been determined
Preconditions	The system has already determined the election winner(s)
Postconditions	The determined election winner(s) have been displayed to the terminal
Main Course	MC1: IR Election 1. The system will display the election type (see UC_4.2.1) 2. The system will display the winner of the election (see UC_08) MC2: CPL Election 1. The system will display the election type (see UC_4.2.1) 2. The system will display the number of seats distributed (//TODO need reference) 3. The system will display the winner candidate(s) of the election (see UC_4.6)
Alternate Course	None
Exceptions	None

4.15 Determine Election Audit File Name

Name	Create an Audit File
ID	UC_4.8
Description	The system will create an audit file containing the history of the election after the votes have all been counted, and a winner has been determined
Actors	Election Officials, Auditors, Testers, Programmers
Organizational Benefits	This will allow an individual to view an audit of the entire election process and can then be downloaded for viewing.
Frequency of Use	Once per election
Triggers	The election has concluded and a winner was determined.
Preconditions	An election has been run and all votes counted. A winner has been determined.
Postconditions	An audit file containing the entire history of the election will be available for viewing in the top level directory

Main Course	1. System creates an Audit file 2. System prompts user that an Audit file has been created
Alternate Course	None
Exceptions	None

4.16 Produce IR Election Audit File

Name	Produce IR Election Audit File
ID	UC_4.8
Description	The system will create an audit file containing the history of the election after the votes have all been counted, and a winner has been determined
Actors	Election Officials, Auditors, Testers, Programmers
Organizational Benefits	This will allow an individual to view an audit of the entire election process and can then be downloaded for viewing.
Frequency of Use	Once per election
Triggers	The election has concluded and a winner was determined.
Preconditions	An election has been run and all votes counted. A winner has been determined.
Postconditions	An audit file containing the entire history of the election will be available for viewing in the top level directory
Main Course	3. System creates an Audit file 4. System prompts user that an Audit file has been created
Alternate Course	None
Exceptions	None

4.17 Produce CLP Election Audit File

Name	Produce CLP Election Audit File
ID	UC_4.8

Description	The system will create an audit file containing the history of the election after the votes have all been counted, and a winner has been determined
Actors	Election Officials, Auditors, Testers, Programmers
Organizational Benefits	This will allow an individual to view an audit of the entire election process and can then be downloaded for viewing.
Frequency of Use	Once per election
Triggers	The election has concluded and a winner was determined.
Preconditions	An election has been run and all votes counted. A winner has been determined.
Postconditions	An audit file containing the entire history of the election will be available for viewing in the top level directory
Main Course	5. System creates an Audit file 6. System prompts user that an Audit file has been created
Alternate Course	None
Exceptions	None

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Project waterfall must be able to evaluate the result of an election of 100,000 ballots in less than 4 minutes. Project waterfall must also be able to be run multiple times during the year during normal election times and or special elections.

5.2 Safety Requirements

Project Waterfall has no safety requirements.

5.3 Security Requirements

Security risks such as ensuring that there is only one vote for one person is handled at voting centers and not through the Voting System Software . The system, through automated file parsing, reduces the need for election officials to open files containing ballot information which may minimize the possibility of file tampering; although, it does not ensure that pre-processed files have not been altered. That is, the system has no security requirements.

5.4 Software Quality Attributes

Project Waterfall is designed to have high usability due to its performance as well as simple interface that election officials, auditors, programmers, and testers can all use simply.

5.5 Business Rules

All actors can use the functionality of the Project Waterfall without restriction.

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

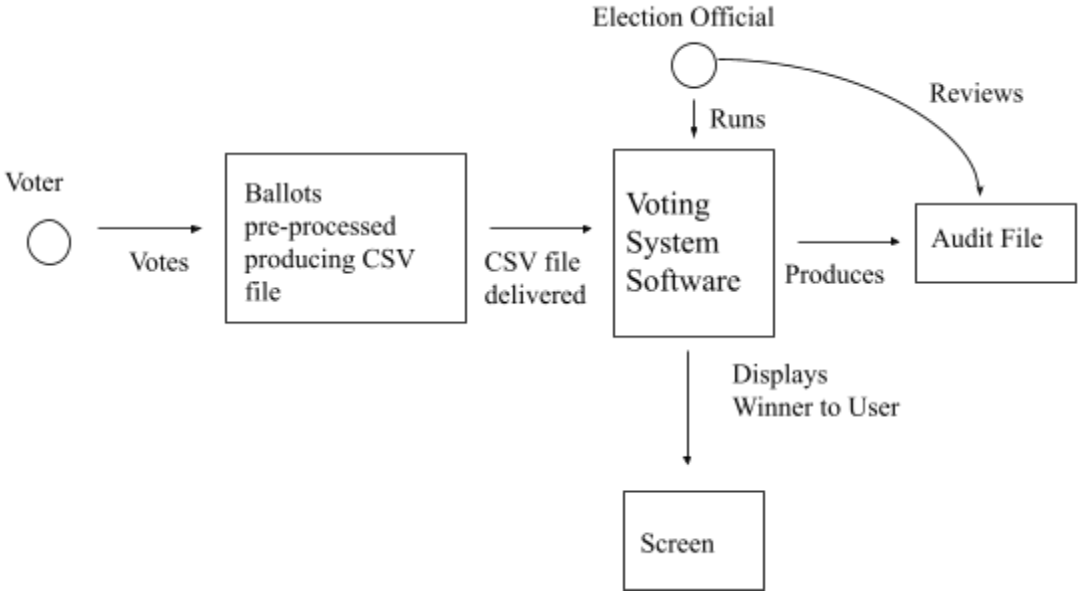
Term	Definition
.csv file	a .csv file is a file with comma-separated values
Audit file	An Audit file will be the file that is produced at the end of an election. (contents of the file is described in UC_4.8)
Auditor	An Auditor refers to a person that conducts an Audit
Ballot	A ballot is the formal document that contains a voters vote(s)
Candidate	A candidate is a person that is running for election.
Candidate Party	A candidate's party is the political party that a

Term	Definition
	candidate is associated with and running under.
Closed Party List Voting (CPL)	On the ballot, voters indicate their preference for a particular party and the parties then receive seats in proportion to their share of the vote
Coin Flip	A coin flip is a random pick chosen fairly that will be used to determine the winner of an election given that there is a tie.
Election	An election refers to a formal and organized choice by vote of a person for a political office or other position.
Election Audit	An election audit is any review conducted after polls close for the purpose of determining whether the votes were counted accurately.
Election Official	Election Official will refer to the person that is responsible for running an election using this system
Instant Runoff Voting (IRV)	all candidates are listed on the ballot. But instead of voting for only one candidate, voters rank the candidates in the order of their preference.
Largest remainder	After the first allocation of seats is complete than the remaining numbers for the parties are compared and the parties with the largest remainders are allocated the remaining seats
Majority	A majority refers to a candidate receiving more than 50% of the votes.
MC	Reference to Main Course
Plurality/Majority Voting	Each Voter is only allowed to vote and count for one candidate. Whichever Candidate that obtains the most votes is elected.
Programmer	A programmer refers to the person(s) responsible for developing the system
Quota	The quota is the number of votes a candidate

Term	Definition
	needs to be certain of election
System	The term system will refer to the Voting System that this document describes.
Tester	A tester will be a person(s) that is responsible for testing the system and ensuring that it is working properly.
Vote	A vote is a choice between two or more candidates
Voter	A person that cast a vote in an election
Write-In Candidate	A candidate who is written on the ballot by the voter to be voted for or ranked by said voter

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>



delete later