Software Requirements Specification

for

Waterfall Election Evaluation

Version 1.2 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	Adding more requirements	1.1
Initial Version	2/16	Adding more requirements	1.2

1. Introduction

1.1 Purpose

The purpose of this document is to present an in-depth description of the Waterfall Election Evaluation. It will explain the purpose and features of the software, what the Waterfall Election Evaluation 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**–those who intend to run 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 Waterfall Election Evaluation is designed to determine election results for either Instant Runoff (IR) or Closed Party List (CPL) elections. It is expected to run multiple times during the year at normal elections time and during special elections. This system will be designed to be fair in its election determinations. The benefits of the Waterfall Election Evaluation are that it allows multiple election types to be processed through a single program and also removes manual processes by automating the counting of votes and determining the 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. Its purpose is to instill trust in the system through transparency and legitimize the results of the election. The system, through automated file parsing, minimizes the possibility of file tampering. The long-term goal of the Waterfall Election Evaluation 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

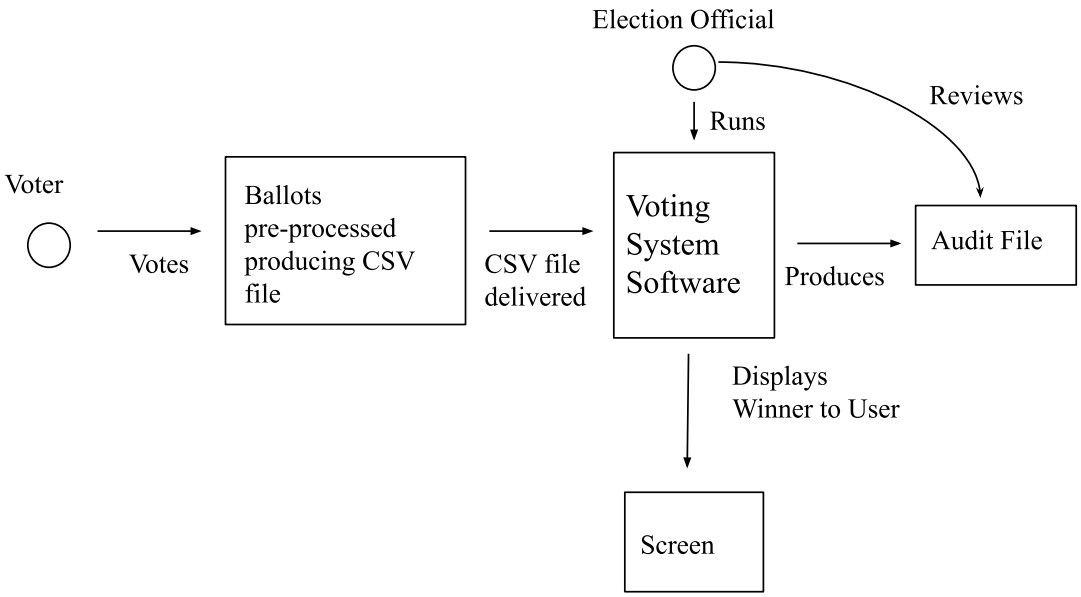

Figure 1: System Interactions

Waterfall Election Evaluation is a 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 all accounted for. The actual voting process is done separately and all ballots are preprocessed before the Waterfall Election Evaluation runs. The Waterfall Election Evaluation reads in the pre-processed comma-delimited (CSV) file that contains information about the casted ballots; then, from the file, the Waterfall Election Evaluation determines the type of election specified and determines 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 will also produce an audit file with election information that contains the winners, a detailed history showing who got what ballot (and order of ballots 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, and 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 determine the type of election based on the contents of the previously read file
- Determine Candidates: the system will determine the candidates needed for an IR election or a CPL
- Determine Ballots: the system will determine the number of ballots in an election
- Determine Seats: the system will determine the number of seats in an election
- Run Election: the system will run an election for either IR or CPL type election
- Run Popularity: the system will run a popularity election if no clear majority in an IR election.
- Fair Coin Toss: the system will flip a coin to determine a winner if there is a tie.
- Display Results: the system will display a brief description of the election after a winner is determined
- 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 Waterfall Election Evaluation 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. Election Officials may choose to share the results of the election with media personnel.
- **Programmers and Testers** who use this product should be familiar with java programming and terminal commands 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 run on a Linux operating system, specifically version 20.03 of Ubuntu. The system will properly run on Dell Precision Computers (the most up-to-date CSE machines) and written in Java using Eclipse version 4.23. The system can be run through the Linux operating systems command prompt.

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 will process CSV files of a specified format and must process the file in that specific format (See section 6). That is, the file structure outside the program is predetermined and can not be changed.

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 help understand how to use the system.

2.7 Assumptions and Dependencies

It is assumed that the system is run on up-to-date Linux systems. It is assumed that all ballots are preprocessed and that they will be viewed as a comma-separated value (.csv) file where each row is separated by a new line. It is assumed that all preprocessing of the file will be done before being placed in the directory. It is assumed that files containing information about IR and CPL elections will follow their respective heading conventions (See section 6). It is assumed that no more than one file will be given to the system. In CPL, it is assumed that all independent groups will only have a single member, and differing independent groups will have different names. For IR, it is assumed that all ballots will have at least 1 person ranked. For CPL, it is assumed that a candidate will be given only 1 ranking. It is assumed that all security concerns are dealt with outside the system. It is assumed that there will be no write-ins for IR or CPL. It is assumed that there are no numbering mistakes in the file–voters will not make any mistakes on the ballot. That is, there are no errors in the ballot. It is assumed that all IR election files follow the specified format (see format in section 6.1). It is assumed that all CPL election files follow specified format (see format in section 6.2).

3. External Interface Requirements

3.1 User Interfaces

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

3.2 Hardware Interfaces

The system will interact directly with the hardware by displaying text and command prompts on the screen. The system shall also function properly with up-to-date versions of Dell Precision Computers.

3.3 Software Interfaces

The system will interact with Linux operating systems, specifically Ubuntu version 20.03, and Eclipse version 4.23. The system will accept an excel (.csv) file that will be read from the same working directory the system is in. The system shall produce an audit (.txt) file to store it within the same working directory.

3.4 Communications Interfaces

The system will not communicate with any outside networks. All ballot files (.csv) will be self-contained in the current working directory. The system does not have any internet communication factors.

4. System Features

4.1 Read in Election File

Name	Read in Election File
ID	UC_4.1
Description	A CSV file (see 6.1 and 6.2) 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, Tester
Organizational	Eases how Election Officials and Programmers can read in a filename

Benefits	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 CSV file There is only one election file to be processed Election file cannot be modified before 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. The filename will be read as an argument from command line 2. Failure of file name read in will result in (EX1) 3. Determine if election parameters will be supplied from the supplied file or through the terminal (go to UC_4.2) 4. Determine election type (go to UC_4.3) 5. If the election type is an IR election (AC2) 6. If the election type is a CPL election (AC3)
Alternate Course	AC1 User Runs Program with No File Input 1. The program prompts the user for filename 2. Failure of file name read in will result in (EX1) 3. Determine if election parameters will be supplied from the supplied file or through the terminal (go to UC_4.2) 4. Determine election type (go to UC_4.3) 5. If the election type is an IR election (AC2) 6. If the election type is a CPL election (AC3) AC2 Election is Determined to be an IR Election 1. Determine candidates (go to UC_4.4) 2. Determine the number of ballots (go to UC_4.6) 3. Read IR ballots (go to UC_4.7) AC3 Election is Determined to be a CPL election 1. Determine party and candidates (go to UC_4.5) 2. Determine the number of seats to be distributed (go to UC_4.7) 3. Determine the number of ballots (go to UC_4.6) 4. Read CPL ballots (go to UC_4.8)
Exceptions	EX1 Invalid Input from the command line or prompt

	1. Display error to user that invalid file name was provided 2. Attempt retrieval of file name again (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, 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 name
Preconditions	The file to be inputted exists in the same directory The file is a CSV file
Postconditions	The system will know whether to prompt the user for input for election parameters or to read all parameters from a supplied file
Main Course	1. Users 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)
Alternate Course	AC1: File does not contain all required information in the file 1. User prompted for any information not in the file. 2. User inputs an7 parameter needed
Exceptions	EX1 Input Type Input 1. User is prompted with an error message that type of inputting election parameters given was invalid 2. Attempt retrieval of input type again (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 (see 6.1 and 6.2)
Actors	Election Official, Tester, 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	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
Main Course	1. File reads first line from CSV file 2. Check if either IR or CPL election
Alternate Course	AC1: Manual Input 1. User gives election type through prompt
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. Attempt retrieval of election type again (AC1)

4.4 Determine Candidates IR

Name	Determine Candidates for IR Election
ID	UC_4.4
Description	Reads in or prompts the user for the candidates in IR that are being voted for (see 6.1)
Actors	Election Official, Tester, Programmer

Organizational Benefits	Allows the system to easily 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 IR 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. Attempt retrieval of IR candidates again (AC1)

4.5 Determine Parties and Candidates CPL

Name	Determine Parties for CPL
ID	UC_4.5
Description	Reads in or prompts the user for the parties in CPL that are being voted for (see 6.1). Independent candidates will belong to groups consisting of only that candidate.
Actors	Election Official, Tester, 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 names of all parties are stored in the system. The names of all candidates for each party are correctly ordered and stored in the system.
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. Attempt retrieval of CPLcandidates again (AC1)

4.6 Determine Number of Ballots

Name	Determine Number of Ballots
ID	UC_4.6
Description	Reads in or prompts the user for the number of ballots in the election file (see 6.1 and 6.2)
Actors	Election Official, Tester, 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	Election file has been read into the system
Preconditions	Determined election type Determined candidates or parties

Postconditions	Number of ballots are stored in the system
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. Attempt retrieval of number of ballots again (AC1)

4.7 Determine Number of Seats

Name	Determine Number of Seats
ID	UC_4.7
Description	Reads in or prompts the user for the number of seats in the election file (see 6.2)
Actors	Election Official, Tester, Programmer
Organizational Benefits	Allows Election Officials to see if the number of seats 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	Election file has been read into the system
Preconditions	Election type is a CPL election
Postconditions	Number of seats are stored in the system
Main Course	1. Reads in the number of seats from file from line 4 + the number of parties 2. Program displays ballot count to the user
Alternate Course	AC1: Manual Input 1. User manually inputs the number of seats to be read from the election file
Exceptions	EX1: Invalid Input

	1. Display error message if user enters invalid input 2. Attempt retrieval of number of seats again (AC1)

4.8 Read IR Ballots

Name	Read IR Ballots
ID	UC_4.8
Description	Read the ballots from the election file. Ballots will be provided in a csv format after the header, with each line representing one ballot (see 6.1). 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, Programmer
Organizational Benefits	Easily allows the system to effectively read the IR ballots in an IR election
Frequency of Use	Once per Election
Triggers	Number of ballots determined
Preconditions	IR election type confirmed Candidates and parties are known
Postconditions	IR ballots are stored in the system
Main Course	Read in the IR ballots from the election file and store it in the system
Alternate Course	None
Exceptions	None

4.9 Read CPL Ballots

Name	Read CPL Ballots

ID	UC_4.9
Description	Read the ballots from the election file. Ballots will be provided in a csv format after the header, with each line representing one ballot (see 6.1). 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, Programmer
Organizational Benefits	Easily allows the system to effectively read the CPL ballots in a CPL 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	CPL ballots are stored in the system
Main Course	Read in the CPL ballots from the election file and store it in the system
Alternate Course	None
Exceptions	None

4.10 Shuffle Ballots

Name	Shuffle Ballots
ID	UC_4.10
Description	Shuffle the order of the ballots
Actors	Election Official, Tester, 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	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	1. Shuffle the order of the ballots in the system
Alternate Course	None
Exceptions	None

4.11 Run IR Election

Name	Run IR Election
ID	UC_4.11
Description	The system will run the election using the rules involved in an IR election See glossary for running of IR algorithm
Actors	Election Official, Programmer, Tester
Organizational Benefits	Runs the series of steps to tally up votes in an IR election to automate the process of determining the winners of an IR election.
Frequency of Use	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 2. Set votes per candidate to first rank votes 3. If a candidate has a majority of the votes, that candidate is declare the winner, IR algorithm is complete

	4. If there there is a tie for the least amount of votes, run a coin flip (UC_4.13) to determine the candidate to eliminate, otherwise eliminate the candidate with the least amount of votes 5. Set votes per candidate to previous votes plus votes for next rank 6. Repeat loop of IR algorithm (MC step 3)
Alternate Course	AC: Election has not been won 1. Eliminate candidate with the fewest votes 2. Retally votes, go to main course step one
Exceptions	EX: There is no majority in IR and only two candidates 1. Run popularity (see UC_4.18)

4.12 Run CPL Election

Name	Run CPL Election
ID	UC_4.12
Description	The system will run the election using the rules involved in an CPL election. See glossary for running of CPL algorithm Seats are distributed to each party in specific order as they were read in from the file. The first candidate will receive the first seat allocated to that party. The second will receive the second seat allocated and so on.
Actors	Election Official, Programmer, Tester
Organizational Benefits	Runs the series of steps to tally up votes in an CPL election and automates the process of determining the winners of an CPL election.
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 per party has been determined

Main Course	1. Tally votes per party 2. Determine quota 3. Determine number of seats allocated based on quota 4. If any party or parties is allocated more seats in the first allocation than they have candidates (AC2) 5. If there are no remaining seats to allocate, allocate seats for both first and second rounds to each party. Candidates are marked as winners based on the amount of seats allocated to their party. CPL algorithm is complete 6. Order remaining votes for parties in descending order 7. If there is a tie between potential parties for seat allocation in second allocation (AC1) 8. If party with most amount of remaining votes does not have anymore candidates to give a seat to (AC3) 9. Allocate seat to party with most amount of remaining seats or winner of tie 10. Eliminate that party's remaining votes 11. Repeat allocation of seats (MC step 6) 12. After seats are allocated to parties. Determine the winner based on the number of seats and order of candidates.
Alternate Course	AC1: Allocate Seat for Tie 1. Determine winner of seat allocation via coin flip (go to UC_4.13) 2. Continue running CLP algorithm (MC step 9) AC2: Party or Parties Do Not Have Enough Candidates for First Allocation 1. Party or parties are allocated seats for all of their candidates 2. Remainder of votes for party is set to 0 3. Continue running CLP algorithm (MC step 5) AC3: Party Does Not Have Enough Candidates for Second Allocation 1. Remainder of votes for party is set to 0 2. Continue running second allocation (MC step 5)
Exceptions	None

4.13 Fair Coin Toss

Name	Fair Coin Toss
ID	UC_4.13
Description	Determines winner from a fair coin toss

Actors	Election Official, Programmer, Tester
Organizational Benefits	Ensures that the election process is fair and instills trust and integrity in the system.
Frequency of Use	Multiple times per election
Triggers	A tie has been determined during the running of an election algorithm. A tie has occurred during IR where two or more candidates have the same lowest number of votes. A tie has occurred during IR where the last two candidates have the same number of votes A tie has occurred in CPL where two or more candidates have the same number of remaining votes A tie has occurred in CPL where there are leftover seats after seats have been allocated for the first pass and remainder delegation.
Preconditions	A tie of votes has occurred between two or more candidates or parties The number of votes per party/candidate are known
Postconditions	One candidate or party is determined to break the tie
Main Course	MC1: IR Election Tie 1. Run random generator 1000 cycles to eliminate bias in seed 2. Assign random number to each tied candidate 3. Generate a random winning number 4. Candidate that has number closest to winning number is declared the winner MC2: CPL Election Tie 1. Run random generator 1000 cycles to eliminate bias in seed 2. Assign random number to each tied party 3. Generate a random winning number 4. Party that has number closest to winning number is declared the winner MC3: IR Candidate with Fewest Votes Tie 1. Run random generator 1000 cycles to eliminate bias in seed 2. Assign random number to each candidate 3. Generate a random number that corresponds to a candidate to simulate a fair a coin toss) 4. Candidate that has number closest to winning number is

	declared the loser
Alternate Course	None
Exceptions	None

4.14 Display Election Results to Terminal

Name	Display Winner of Election
ID	UC_4.14
Description	Display election information of the election after a winner or winners have been determined Election information for an IR election should include type of the election Election information for a CPL election should include type of the election as well as the number of seats distributed
Actors	Elections Official
Organizational Benefits	Allows for easy viewing of a winner after an election has been run
Frequency of Use	Once per election
Triggers	Election algorithm has completed evaluation of election
Preconditions	Election election winner(s) have been determined
Postconditions	The determined election information and election winner(s) have been displayed to the terminal
Main Course	MC1: IR Election 1. The system will display the election type (see UC_4.3) 2. The system will display the winner of the election (see UC_4.10) 3. The number of ballots cast, stats–such as number of votes received, and percentage of votes received will display MC2: CPL Election 1. The system will display the election type (see UC_4.3) 2. The system will display the number of seats distributed (see UC_4.5)

	3. The system will display the winning candidate(s) of the election (see UC_4.11) 4. The number of ballots cast, stats–such as number of votes received, and percentage of votes received will display
Alternate Course	None
Exceptions	None

4.15 Determine Election Audit File Name

Name	Determine Election Audit File Name
ID	UC_4.15
Description	Determines audit file name. The file name will be based on the name of the election type and date of the election. For instance, if the election was IR and run on 2/16/23, then the audit file name would be “IR_2/16/23.txt”.
Actors	Election Officials, 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 or winners were determined
Preconditions	Election filename is stored in the system
Postconditions	Audit file name is stored in the system
Main Course	1. Audit file name is generated from the election type and date that the audit file is generated 2. Audit file name is stored in the system
Alternate Course	None
Exceptions	None

4.16 Produce IR Election Audit File

Name	Produce IR Election Audit File
ID	UC_4.16
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 (see 6.3)
Actors	Election Officials, 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 Audit file name 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. The system creates an audit file with the determined name 2. Audit file is generated with the election information at the time (e.g. Type of Voting, Number of Candidates, Candidates, Number of Ballots, calculations, how many votes a candidate had, etc) 3. The audit file will show the order of ballots being assigned to the candidates 4. The audit file will show the order of removal of the candidates in IR and what ballots were redistributed and show all steps.
Alternate Course	None
Exceptions	None

4.17 Produce CLP Election Audit File

Name	Produce CLP Election Audit File
ID	UC_4.17

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 (see 6.4)
Actors	Election Officials, 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 with determined name 2. Audit file is generated with the election information at the time (e.g. Type of Voting, Number of Candidates, Candidates, Number of Ballots, calculations, how many votes a candidate had, etc) 3. The audit file shows which parties got what ballots
Alternate Course	None
Exceptions	None

4.18 Run Popularity

Name	Run Popularity Election
ID	UC_4.18
Description	The system will run the election using the rules involved in popularity See glossary for running of popularity
Actors	Election Official, Programmer, Tester
Organizational Benefits	Determines a winner in the case that no party has a majority after all votes have been reallocated and only to candidates remain
Frequency of Use	Once per IR election with two remaining candidates
Triggers	There is no clear majority in IR after all votes have been handed out

Preconditions	Election type is IR Number of votes for last two candidates is known All votes have been handed out
Postconditions	IR election winner has been determined
Main Course	1. Choose candidate with highest number of votes 2. Display winner and results (see UC_4.14)
Alternate Course	AC: Election has a tie 1. Do coin toss (see UC_4.13)
Exceptions	None

5. Other Nonfunctional Requirements

5.1 Performance Requirements

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

5.2 Safety Requirements

The Waterfall Voting Evaluation system 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 Waterfall Election Evaluation. That is, the system has no security requirements.

5.4 Software Quality Attributes

The Waterfall Voting Evaluation system is designed to have high usability due to its performance as well as the simple interface that election officials, programmers, and testers can all use.

5.5 Business Rules

All actors can use the functionality of the Waterfall Voting Evaluation system without restriction.

6. Other Requirements

6.1 IR Election File Format

File will be of type .csv
Header
Line 1 : Election Type (IR for Instant Runoff)
Line 2 : Number of Candidates
Line 3 : Candidates Separated by Commas
Line 4 : Number of Ballots
Line 5+: Ballots

```
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
```

Figure 2: Example of IR election file

6.2 CPL Election File Format

File will be of type .csv
Header
Line 1 : Election Type (CPL for Closed Party Listing)
Line 2 : Number of Parties
Line 3: Parties Separated by Commas
Line 4 + number of parties: Name of candidates of the parties separated by commas
Line after candidates: Number of seats
Next line: Number of ballots
Remaining Lines: Ballots

```
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,,,,
```

Figure 3: example of CPL election file

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

Term	Definition
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 candidate is associated with and running under.
Closed Party List Voting (CPL)	Voters indicate their preference for a particular party on their ballot and the parties then receive seats in approximate proportion to their share of the total vote. Algorithm to determine seat allocations: The quota is calculated (see quota) For the first round of seat allocations, seats are allocated by whole numbers chunks of the quota size of each party's votes. For example, if the quota is 10,000 and a party receives 32,000 votes, it will receive 3 seat allocations for the first round. For the second round of seat allocations, the votes that did not contribute to the first round seat allocation are considered. Considering the example from the first round, the votes considered will be 2,000 votes. The seats not allocated in the first round will be allocated to the parties in descending order of remaining votes after the first allocation.

Term	Definition
	The party will receive seats that are a sum of those received from the first and second rounds of allocations. The party will then choose to elect candidates that are sorted in a list of priority to be elected. Those with a higher priority will receive an allocated party seat over those with a lower priority. That is, the candidates have a particular order in which seats are allocated if their party wins.
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)	Voters can rank all the candidates on their ballot based on their preference of who they want to win the election. These rankings range from 1 to the number of candidates. Algorithm to determine winner: The candidates' votes are set to be the number

Term	Definition
	of first place votes they each received. If no candidate holds the majority of the vote, the candidate with the least amount of votes is eliminated as a potential winner. The votes for the second rank is then added to the votes of each candidate. If no candidate holds a majority, another candidate is eliminated in a similar manner and this process continues until a candidate reaches a majority of the votes. If a candidate ever reaches a majority of the votes, they are declared the winner If a candidate never reaches a majority after all the rank votes are distributed, the candidate with the plurality wins.
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.

Term	Definition
Programmer	A programmer refers to the person(s) responsible for developing the system
Quota	The quota is the result of the total number of votes divided by the number of seats to be allocated in a CPL election
System	The term system will refer to the Waterfall Election Evaluation software 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