

Name	Input file name through prompt	Input information separate from file given	Collecting information from file about the election	Producing the audit file
ID	UC_010	UC_011	UC_012	UC_013
Description	User enters a file name through the terminal when running the voting simulation.	User provides information manually through the terminal in addition to or in place of a file name.	Ballot data is extracted from the input file.	The audit file is created and saved.
Actors	Election official user	Election official user	System/machine	System/machine
Organizational Benefits	Starts simulated voting process for collected ballots saved in file.	Allows faster processing of ballots.	Reduces time spent counting ballots.	Provides important data about the election and its progression.
Frequency of Use	Every time product is used.	Prompted during each use.	Every time a file is used as input.	Every time program is used.
Triggers	The program is initiated.	The user selects to manually input information.	A file is selected as input by the user.	Triggers once program is provided input.
Preconditions	The user starts the program.	The user has input a file.	The user has input a file.	The user provides all necessary input.
Postconditions	The program notifies the user that the file was successfully read.	The program notifies the user that is has aquired the additional information	The information from the file is saved by the program and available to the system.	The file is available to the user in the "audits" folder of the program directory.
Main Course	1. The user starts the program. 2. The system prompts user for file input. 3. User types a file name into the terminal. 4. System opens and reads the file (see EX1, 2, UC_012).	1. System prompts user to choose to input additional information. 2. User selects the type of information to input. 3. System prompts the user for input. 4. User inputs the appropriate data. 5. The system saves the input.	1. System opens the file. 2. System extracts all information about the election. 3. Information is saved and used by the program.	1. Audit file entity is created by the system. 2. All appropriate data is appended to the file entity. 3. Audit text file is saved to the "audits" folder of the program directory.
Alternate Courses	No alternative courses.	AC1 User chooses not to input additional information: 1. Skip Main Course 2-5.	No alternative courses.	No alternative courses.
Exceptions	EX1 The system is unable to read or open the file: 1. System notifies the user of the error. 2. System returns to Main Course step 2.	EX1 The input data is inconsistent with the file data: 1. System notifies user of the error. 2. System returns to Main Course step 1.	EX1 The system is unable to extract data from the file: 1. System notifies user of the error. 2. System asks if user wants to retry or select another file.	EX1 The system is unable to save the file: 1. System notifies user of the error. 2. System prompts user to select a location to save the file.

Counting ballots through ranking for each candidate in IR voting	Counting popularity of party through ballots in party list voting.	Calculating winner in IR voting - no tie	Calculating winner in party list voting - no tie	Redistributing votes and removing least popular candidate in IR voting
UC_014	UC_015	UC_016	UC_017	UC_018
Ballots are counted for an instant runoff election.	Ballots are counted for a closed party list election	The winner of the election is determined in an instant runoff election.	The winner of the election is determined in a closed party list election.	When there is no majority on voter preference, the candidate with the least voting preference is removed and votes are redistributed.
Election official user	Election official user	Election official user	Election official user	Election official user
Allows users to count the ballots of an IR election.	Allows users to count the ballots of a closed party list election.	Allows election officials to quickly determine the winner of an instant runoff election.	Allows election officials to quickly determine the winners of a closed party list election.	Allows elections to slim down candidate options to more effectively decide a winning candidate.
Appx. half of all uses.	Appx. half of all uses.	Appx. half of all uses.	Appx. half of all uses.	Appx. half of all uses.
An input file is designated as IR on the first line.	An input file is designated as CPL on the first line.	Ballot counting has completed and the data is available to the system	Ballot counting is completed and the data is available to the system.	There is no candidate with a majority preference of over 50%
The user provides all necessary input.	The user provides all necessary input.	Ballot counts are available in the system.	Ballot counts are available in the system.	There is no clear winner in the election
Ballot data is available for calculations of what candidate will be eliminated.	Ballot data is available for calculations of seat allocations for each party.	Election winner is available to the system.	Election winner is available to the system.	Votes are distributed accordingly and used for further calculations
1. System creates data structure to keep track of votes for each candidate. 2. System traverses list of ballots and counts appropriate preference votes for each candidate, updating number of votes for each candidate.	1. System creates data structure to keep track of votes for each party. 2. System traverses list of ballots and counts votes for each party.	1. System counts first preference votes. 2. System eliminates candidate with least amount of votes (UC_018). 3. System checks whether a candidate has a majority vote and, if so, declares the winner. 4. System distributes the eliminated candidate's second preference votes to the remaining candidates. 5. Repeats from step 1.	1. System determines the quota by dividing the total number of votes by the number of seats up for election. 2. System counts the votes for each party. 3. System divides the votes for each party by the quota to determine the first allocation of seats. 4. System removes (seats won * quota) number of votes from each party's total votes. 5. System awards the remaining seats to the parties with the largest remainders.	1. Calculations are done for current candidates to determine if there a 50% preference voting for one candidate through rankings. 2. System removes using candidate from some data structure by calculating which candidate has the lowest voting preference. 3. Votes are redistributed through ballots who chose the removed candidate for their first preference and thus will gather their second preferred candidates and add to those candidates accordingly.
No alternative courses.	No alternative courses.	No alternative courses.	No alternative courses.	AC1 If there is a tie when determining the lowest candidate, go to UC_019 before returning to Main Course Step 3
No exceptions are expected.	No exceptions are expected.	EX1 There is a tie between the 2 lowest ranked candidates: 1. Determine a winner of the tie (UC_019,020). 2. Continue from Main Course step 4.	No exceptions are expected.	EX1 There is only one candidate, they will be automatically the winner of the election.

Determining a winner when there is a tie between two candidates	Determining a winner when there is a tie between more than two candidates	Allocating seats for each party and determining winning candidates from list	Displaying results and information of election to interface
UC_019	UC_020	UC_021	UC_022
When there are two candidates with the same amount of voting preference in ballots, a coin-flip will determine the winner	When there are more than two candidates with the same amount of voting preference in ballots, a coin-flip will determine the winner	In a closed party list election, there will be candidates selected as winners based on the number of seats won by each party.	Statistics about the election and its winners are output to the interface.
Election official user	Election official user	Election official user	Election official user
Creates a fair system where candidates with equal popularity have the same chance of winning 50% percent of the time	Creates a fair system where candidates with equal popularity have the same chance of winning a specific percentage of the time	Allows election officials to easily see the candidates elected to the seats.	Provides election officials access to the results and statistics of the election as soon as they are calculated.
Appx. a few of all uses	Appx. very few of all uses	Appx. half of all uses	Every use
Two candidates are determined to have the same amount of voting preference	More than two candidates are determined to have the same amount of voting preference	The seats up for election in a CPL election have been assigned to the winning parties.	The program has run to completion and winners have been determined.
There are two candidates with calculated voting preference	There are more than two candidates with calculated voting preference.	Ballot counts and winning parties data is available in the system.	Audit data is available for retrieval by the system.
A winner is determined out of the two candidates	A winner is determined out of the candidates.	The winning candidates for each party are designated.	Data is available to the user through the interface.
1. Two candidates are passed who have the same amount of voting preference 2. A coin-flip giving each candidate 50% to win is performed resulting in one candidate being the output	1. The candidates are passed who have the same number of votes. 2. The system randomly chooses a candidate to eliminate and returns to the winner calculation	1. System retrieves number of seats won by each party. 2. System designates the winning candidates based on party and seats won. 3. System saves data for display and audit production.	1. System retrieves all appropriate election data from either the audit data that was produced as the program progressed, or other internal sources. 2. System displays the data to the interface in a structured format.
No alternative courses.	No alternative courses.	No alternative courses.	No alternative courses.
No exceptions are expected.	No exceptions are expected.	No exceptions are expected.	No exceptions are expected.