Name	Opening file.
ID	UC_001
Description	Someone sends a file to an election official and we open the file.
Actors	Election Official
Organizational Benefits	This is needed in order to read in election results using the voting system.
Frequency of Use	Every time the program is run.
Triggers	The file is sent to the election official to validate.
Preconditions	1. CSV file exists. 2. File is in the same directory as the program. 3. File is readable.
Postconditions	1. File is open and ready to be read and processed by the system.
Main Course	1. Prompt user for the full file name. 2. Attempt to open the file. (see EX1)
Alternate Courses	None.
Exceptions	EX1 Incorrect file name: 1. Reprompt the user for the filename. 2. Return to MC2.

Name	Process header
ID	UC_002
Description	We need to be able to process the header of the voting data to determine the number of candidates, candidate names, type of voting algorithm, and other information specific to each algorithm.
Actors	Election Official
Organizational Benefits	This is needed in order to get the information needed to determine the appropriate algorithm and process ballots.
Frequency of Use	Every time the program is run.
Triggers	The file has been opened by the program.
Preconditions	1. Election data file has been opened by the program successfully. 2. File is in the correct and valid format.
Postconditions	1. Number of candidates / names of candidates are available to the program. 2. Type of voting algorithm to be used has been determined. 3. Number of ballots is stored and available for the algorithm to use. 4. If the algorithm is OPL, the number of parties is stored and available for the program.
Main Course	1. Read the type of voting system from the first line of the header. 2. Read the number of candidates and store it. 3. Read the list of candidates and store it. 4. Read the next line containing the number of ballots in IR voting (see AC1).
Alternate Courses	AC1: Voting system is OPL 1. Read the next line containing the number of parties and store it. 2. Read the next line containing the number of ballots and store it.
Exceptions	None

Name	Create an Audit File
ID	UC_003
Description	After the user inputs the filename, an audit file is created with the order of ballots assigned to candidates. The file shows the order of how candidates were removed in IR and the ballots that were redistributed.
Actors	Election Officials
Organizational Benefits	This is needed to communicate the distribution of ballots and which candidates were removed.
Frequency of Use	Every time the program is run.
Triggers	User inputs .csv file name.
Preconditions	1. Ballots were assigned to candidates. 2. Some candidates were removed (IR). 3. Ballots redistributed and winner found.
Postconditions	1. File indicates the ranking of candidates and how ballots were redistributed after each instant runoff. 2. File shows type of voting, number of candidates, candidates, number of ballots, and calculations. 3. Denotes winner.
Main Course	1. For each round of voting, the redistribution of ballots is shown for IR, or the assignment of seats to parties and candidates is shown for OPL. 2. Information is written to the file.
Alternate Courses	None
Exceptions	None

Name	Display Election Results
ID	UC_004
Description	The program displays the winner of the election when run.
Actors	Election official
Organizational Benefits	Allows the election official to see the result of the election quickly and effectively.
Frequency of Use	Everytime the program is run.
Triggers	The specified algorithm is completed, and the winner(s) is found
Preconditions	1. The program has opened the CSV file and processed the header data, including election type, and the ballot information. 2. The selected algorithm has been completed on the data 3. Election winner(s) has been found
Postconditions	1. The election official's screen displays the name of the winning candidate, the type of the election run, the number of seats, number of ballots cast, and number of votes for each candidate.
Main Course	1. Print out type of election ran (if OPL election: see UC_007). 2. Print out number of seats filled. 3. Print out number of ballots cast. 4. Print out number of votes for each candidate. 5. Print out winner(s) of election.
Alternate Courses	None
Exceptions	None

Name	Determine Election Tie
ID	UC-005
Description	When there is a tie, a fair tie breaker must be done by coin toss simulation to declare the winner.
Actors	Developer and Tester
Organizational Benefits	Will prevent elections from not fairly determining the winner(s) and being called again.
Frequency of Use	Whenever there is a tie in the final number of votes for 2+ candidates.
Triggers	There is a tie between 2+ candidates.
Preconditions	1. Ballots have been completely counted. 2. If an IR election, the amount of previous candidates' votes for each candidate have been applied.
Postconditions	1. A candidate is selected.
Main Course	1. Assign a number from 0-(n-1) for each candidate, where n is the total number of candidates. 2. Generate a random number. 3. Divide by the number of candidates and keep the remainder. 4. Should the remainder match the candidate, then that candidate is the selected candidate.
Alternate Courses	None
Exceptions	None

Name	No clear majority in IR
ID	UC_006
Description	Resolve ties in the IR voting system after all of the votes have been tallied and there are 2 candidates left, and there is still no clear winner.
Actors	Election Official, Developers, and Testers
Organizational Benefits	Allows us to declare a winner in the event of no clear majority, needed to make the election algorithm fair.
Frequency of Use	Any time there is a situation where no candidate has more than 50% of the vote after all of the rounds of IR have been completed.
Triggers	There are two candidates left in an IR election and neither has more than 50% of the vote.
Preconditions	1. IR election file has been read. 2. Election has been processed and the algorithm has reached its conclusion. 3. There are two remaining candidates, neither of which have 50% of the vote.
Postconditions	1. Winner has been declared.
Main Course	1. Determine which of the two candidates have the most votes. 2. Declare the winner based on the candidate with the most votes (see AC1).
Alternate Courses	AC1: Two candidates have the same number of votes: 1. Flip a coin, and declare the winner of the coin toss the winner of the election. See UC_005.
Exceptions	None

Name	Displays the number of seats won by each party.
ID	UC_007
Description	Displays a list of the parties with the majority of seats to those with the least amount of seats resulting from the ballots.
Actors	Developers, Election Officials, and Testers
Organizational Benefits	This is needed to communicate the amount of people from a party that will be in office.
Frequency of Use	Every time the program is run.
Triggers	The filename is submitted by the user with ballot information.
Preconditions	1. Ballots are submitted. 2. Candidates have been removed/reordered. 3. Voting parties were ranked with majorities. 4. File name is submitted. 5. Ballots are processed.
Postconditions	1. Seat numbers are displayed for each party.
Main Course	1. Candidates are reordered and removed 2. Parties are ranked based on the number of seats. 3. Display the number of seats won by each party.
Alternate Courses	None
Exceptions	None

Name	Process Instant Runoff Election
ID	UC_008
Description	The program runs an Instant Runoff algorithm and finds a winner.
Actors	Developers, election officials, and testers
Organizational Benefits	Allows the election officials to use the program to determine the election winner with an instant runoff approach.
Frequency of Use	Every time the program is passed a file with Instant Runoff in its header.
Triggers	The program is passed a file with Instant Runoff in the header.
Preconditions	1. The program has been passed a file and it has been opened. 2. The file has a specifier in its header that says that it's an Instant Runoff ballot. 3. Each ballot has at least one candidate ranked. 4. Voter has ranked anywhere from 1 to the number of candidates, not skipping any ranks. 5. None of the ballots have write-in candidates. 6. Ballots have been processed (UC_011).
Postconditions	1. The program has found an election winner by processing the ballot and running the instant runoff algorithm.
Main Course	1. Record the ranking of all candidates on each ballot. 2. Sum the total votes for each candidate (taking only first ranked on each ballot). 3. Check if any candidate got more than 50% of the total votes cast. 4. If not, the candidate with the fewest votes is eliminated, and all of the ballots that had that candidate as the first ranked candidate, now count towards the candidate that is ranked directly below the now eliminated candidate. (see AC2). 5. Repeat steps 3 and 4 until a candidate gets more than 50% of the total votes cast (see AC1) (see AC3). 6. The winner is the candidate that received the majority of the votes cast.
Alternate Courses	AC1: A tie in number of votes when it gets down to 2 candidates left 1. Begin UC_005. AC2: A tie in the number of votes between the two lowest voted for candidates when a strong majority has not been found yet. 1. Begin UC_005.

	AC3: When only two candidates remain and all ballots have been dispersed, but neither candidate has a true majority 1. The candidate with more votes than the other candidate is the winner of the election.
Exceptions	None

Name	Determine Results of Open Party List Election
ID	UC-009
Description	When an Open Party List election is held, the Open Party List algorithm must be used.
Actors	Developer, Election Officials, and Testers.
Organizational Benefits	Meets requirements of election caller.
Frequency of Use	Everytime the program is passed a file with “OPL” in the header.
Triggers	Open Party List is noted in the header of the input file.
Preconditions	1. User has inputted the file name. 2. The file has been opened. 3. Header specifies Open Party List election. 4. Only one file has been imputed. 5. All independents are grouped into one party. 6. Ballots have been processed (UC_011).
Postconditions	1. Allotted seats per party have been calculated. 2. Winners of each seat have been calculated.
Main Course	1. Divide the total amount of votes by the amount of available seats. Divide the amount of votes for each party by the aforementioned quotient. This is the initial amount of seats allocated to each party. 2. Sum total amount of seats from all parties (See AC1) 3. For each party, select the top m candidates and note them as winners, where m is the amount of seats won for that party. (See AC2)
Alternate Courses	AC1: Sum of total seats is less than the amount of seats available. 1. Find the difference between the amount of seats currently allocated and the number of seats available. 2. Rank each party by the remainder of the division in step 2. Select the top r parties and assign another seat to each party, where r is the difference between the amount of available and allocated seats. 3. Return to MC step 3. AC2: Tie between party members: 1. Begin UC_005. 2. Return to MC step 3.
Exceptions	None

Name	Turn off ballot shuffling
ID	UC_010
Description	Ballot shuffling is an important part of the voting process, but in order to know the expected election results for testing, we need to be able to turn it off.
Actors	Testers
Organizational Benefits	This is needed to control the expected election results so we can verify the voting algorithms are working properly.
Frequency of Use	Every time the program is run in testing mode.
Triggers	A tester runs the program in testing mode.
Preconditions	1. Program has been started, with an additional argument flag “-t” to indicate testing mode.
Postconditions	1. A winner is declared and an audit file is produced.
Main Course	1. The program executes normally, but the ballot shuffling functionality is bypassed to keep the order of the ballots the same. 2. Audit file produced indicates that the election was processed with ballot shuffling turned off.
Alternate Courses	None
Exceptions	None

Name	Process Ballots
ID	UC_011
Description	Process the ballots in the CSV file to count the votes.
Actors	Election officials, developers, and testers.
Organizational Benefits	Processing ballots is the first step in determining an election winner.
Frequency of Use	Every time an election is run through the program.
Triggers	Header has been processed.
Preconditions	1. File name has been imputed and validated. 2. File has been opened. 3. Header has been processed. 4. File is in the correct format. 5. Every ballot has at least one selection.
Postconditions	1. Each ballot has been counted.
Main Course	1. Note the selections made on the ballot. 2. Count the ballot toward the indicated candidate.
Alternate Courses	None
Exceptions	None