Name	Prompt User for Election Metadata
ID	UC_ASK_META
Description	After the file is successfully opened, the user will be given the option to manually input the election metadata (type of election, number of candidates, number of ballots, etc).
Actors	Programmers, system testers, or election officials.
Organizational Benefits	Allows easier and faster processing of election results since the program does not need to process the file to determine this information. Can also improve performance when multiple election files are used (multifile elections are not currently a capability of the system, but it may be in the future).
Frequency of Use	This will occur every time the program is run with a valid .csv file.
Triggers	The program opens the .csv election file successfully.
Preconditions	The program has received a valid .csv file name from the user and has successfully opened that file for reading.
Postconditions	Important election metadata which will be used in the election algorithms has been stored by the system.
Main Course	1. The user is asked if they would like to manually enter metadata. 2. The user types “y”. 3. The user is prompted for the election type and then enters “IRV” for instant runoff voting. 4. The user is asked to input the number of candidates and does so. 5. The user is asked to enter the candidates names followed by the first letter of their party in parentheses. The names with the party is a comma separated list. 6. The user is asked to enter the number of ballots and does so.
Alternate Courses	AC1: The user chooses CPL voting 1. The user is asked if they would like to manually enter metadata. 2. The user types “y”. 3. The user is prompted for the election type and then enters “CPL” for closed party list voting. 4. The user is asked to input the number of parties and does so.

	5. The user is asked to enter the names of the parties as a comma separated list and does so. 6. The user is continually prompted to enter a comma separated list of the candidates for a particular party and does so. 7. The user is asked to input the number of seats and does so. 8. The user is asked to input the number of ballots and does so. AC3: The user chooses not to manually input 1. The user is asked if they would like to manually enter metadata. 2. The user enters "n". 3. Move to main course of UC_PARSE_ET.
Exceptions	EX1: The user does not enter a number when asked. 1. Error message is printed. 2. Return to main course step 1. EX2: The user does not list the same number of candidates/parties as they previously entered. 1. Error message is printed. 2. Return to main course step 1. EX3: The user does not input a valid election type. 1. Error message is printed. 2. Return to main course step 1.

Name	Display IRV election results to terminal.
ID	UC_IRV_RESULTS
Description	Once the voting file has been processed and the IRV voting algorithms have been run on the ballot data, the program will output important information regarding the results of the IRV election.
Actors	Programmers, system testers, or election officials.
Organizational Benefits	Allows a quick summary of election results for election officials. Allows results to be understood in a quick and clear manner. Additionally, this allows results to be quickly and securely passed along to media personnel.
Frequency of Use	This will occur every time the program is run on an election file that uses Instant Runoff Voting.
Triggers	The function responsible for IRV result output is called.
Preconditions	The program has processed the data in the IRV ballot file and has determined the winner based on the IRV election algorithms. All

	other information which will be displayed (see main course) has been determined. Any ties have been handled (see UC_TIE).
Postconditions	Information regarding the results of the election are present on the terminal screen. The program terminates.
Main Course	1. The election type “Instant Runoff Voting” is printed to the terminal. 2. A table is then printed which will show the winner of the election and the vote totals for all candidates. 3. The table will show the candidate names and their party in the left column, with the winner having an asterisk before their name. There will then be a column for the number of votes received and a column for percentage of total votes acquired. 4. The program terminates.
Alternate Courses	AC1: Initially no candidate had a majority; one transfer required to find a winner. 1. The election type “Instant Runoff Voting” is printed to the terminal. 2. A table is printed. The first column shows the candidates and their parties, and the winner has an asterisk before their name. The second column will show the number of original first choice votes and the percentages of the total first choice votes as well. The third column will show the number of votes gained by the three highest vote getters from the last place candidate. The new totals of votes are displayed in the fourth column, as well as the percentage of votes after adjustment. 3. The program terminates. AC2: Two vote transfers were required to find the winner. 1. The election type “Instant Runoff Voting” is printed to the terminal. 2. A table is printed. The first four columns are the same as AC1: step 2, but two additional columns are added here. The fifth column will show the number of votes transferred from the third place candidate to the two top candidates. The fifth column will show the new totals and the percentage of votes for the final two candidates. Note that the winning candidate may or may not have a majority at this point (see use case UC_IRV_POPULARITY_VOTE). 3. The program terminates. AC3: More than two vote transfers were required to find the winner. 1. Execute steps 1-2 of AC2.

	2. Any additional rows will be added to display additional vote transfers. These vote transfers will be displayed the same way that they are described in AC1: and AC2.
Exceptions	EX1: User presses “ctrl+C” to end the program early. 1. Program will terminate, with some output already printed. How much will depend on when the user terminates the program.

Name	Display CPL election results to terminal.
ID	UC_CPL_RESULTS
Description	Once the voting file has been processed and the CPL voting algorithms have been run on the ballot data, the program will output important information regarding the results of the CPL election.
Actors	Programmers, system testers, or election officials.
Organizational Benefits	Allows a quick summary of election results for election officials. Allows results to be understood in a quick and clear manner. Additionally, this allows results to be quickly and securely passed along to media personnel.
Frequency of Use	This will occur every time the program is run on an election file that uses Closed Party List voting.
Triggers	The function responsible for CPL result output is called.
Preconditions	The program has processed the data in the CPL file and has determined the winners based on the CPL election algorithms. All other information which will be displayed (see main course) has been determined. Any ties have been handled (see use case UC_TIE). Any necessary random assignment of seats has been handled (see use case UC_SEAT_LOTTERY_CPL).
Postconditions	Information regarding the results of the election are present on the terminal screen. The program terminates.
Main Course	1. The election type “Closed Party List Voting” is printed to the terminal. 2. A table is then printed which will display election results. 3. The table will contain 7 columns. The first column will display the names of all parties in the election. The second column will show the number of votes that each party

	received. The third column will show the number of seats that each candidate has earned after the initial allocation of seats. The fourth column will show the number of remaining votes for each party after this initial seat allocation. The fifth column will then show how many seats are allocated to each party in the second allocation of seats. The sixth column shows the final number of seats for each party. The seventh column will display the percentage of the vote for each party and also the percentage of the total number of seats that that party has been allocated. 4. The program terminates.
Alternate Courses	No alternate courses.
Exceptions	EX1: user presses “ctrl+C” to end the program early. 3. Program will terminate, with some output already printed. How much will depend on when the user terminates the program.

Name	Get ballot file name from the user.
ID	UC_GET_FILENAME
Description	The user will be prompted to enter the name of the file which contains the election information as well as the data for all ballots in the election.
Actors	Programmers, system testers, or election officials.
Organizational Benefits	Provides a user-friendly way of telling the program which file will need to be processed. If an error is made by the user, the program will handle this and reprompt the user, which will save time as the user will not need to rerun the program.
Frequency of Use	This will occur every time the program is run with no command line argument.
Triggers	The program is run from the terminal with no command line argument.
Preconditions	The program does not have any stored command line arguments when the program begins execution (since none were given).
Postconditions	The program has a valid comma separated values file name that it will then attempt to open (see use case UC_OPEN_BALLOT_FILE).

Main Course	1. The user runs the program from the command line. 2. The program recognizes that no command line arguments were given. 3. The program prompts the user to enter the name of the ballot file (must be a .csv file). 4. The program saves this file name somehow. 5. The program verifies that the file type is correct. 6. The program moves on to attempt opening the file (see use case UC_OPEN_BALLOT_FILE).
Alternate Courses	No alternate courses.
Exceptions	EX1: User enters a filename that is not of type .csv 1. Message is printed to screen informing the user that the file type is invalid and it must be .csv. 2. Return to step 2 of the main course.

Name	Get ballot file name from command line.
ID	UC_CMD_LINE_FILENAME
Description	The program receives and internalizes the name of the election ballot file from a command line argument given by the user.
Actors	Programmers, system testers, or election officials.
Organizational Benefits	Allows a fast way of providing the ballot file name to the program. Allows the user to bypass being prompted and move forward to data processing. Error checking is done to ensure an accurate file name is obtained which will ensure proper election result processing.
Frequency of Use	This will occur every time the program is run with one or more command line arguments.
Triggers	The program is run from the terminal with one or more command line arguments.
Preconditions	The program has one or more command line arguments stored when the program begins execution.
Postconditions	The program has a valid comma separated values file name that it will then attempt to open (see use case UC_OPEN_BALLOT_FILE).
Main Course	1. The user runs the program from the command line with one command line argument. 2. The program recognizes that a command line argument has

	been received. 3. The program verifies that the command line argument is a file name with type .csv. 4. The program moves on to attempt opening the file (see use case UC_OPEN_BALLOT_FILE).
Alternate Courses	AC1: More than one command line argument given. 1. The user runs the program from the command line with more than one command line argument. 2. The program recognizes that command line arguments have been received. 3. The program checks the first command line argument to see if it is a valid .csv file name. 4. The program moves on to attempt opening the file (see use case UC_OPEN_BALLOT_FILE).
Exceptions	EX1: User enters a filename that is not of type .csv 1. Message is printed to screen informing the user that the file type is invalid and it must be .csv. 2. Return to step 2 of the main course of use case UC_GET_FILENAME.

Name	Opening ballot file for reading.
ID	UC_OPEN_BALLOT_FILE
Description	The program opens up the .csv ballot file that was given by the user and prepares to read data from it.
Actors	Programmers, system testers, or election officials.
Organizational Benefits	Ensuring proper opening of the ballot file is crucial for determining the winner of the election, since the information needed to determine the winner is in the file. Proper handling will save time and ensure accuracy.
Frequency of Use	This will occur every time the program is run and has been given a .csv file name from the user.
Triggers	The user enters a valid .csv file name when prompted.
Preconditions	The program has a .csv file name from the user. A .csv file with this name is located in the current working directory.
Postconditions	The .csv file containing the election and ballot information is successfully opened for reading.

Name	Opening ballot file for reading.
ID	UC_OPEN_BALLOT_FILE
Main Course	1. The program attempts to open the file with the given name for reading. 2. The file is successfully opened.
Alternate Courses	No alternate courses.
Exceptions	EX1: The file fails to open because the file does not exist in the current directory. 1. Error message is printed to the screen saying that the file was not opened due to the fact that there is no file with that name in the directory. 2. Return to step 2 in the main course of use case UC_GET_FILENAME.

Name	Make/open audit file
ID	UC_CREATE_AUDIT
Descriptions	To verify the election, an official needs an audit file to confirm the process of the election and the results of the software.
Actors	Election official
Organizational Benefit	Verifiability of the election and software.
Frequency of Use	Every time the software is used.
Triggers	The software is used and election data has been processed.
Preconditions	The .csv file has been processed, and the election results have been produced.
Postconditions	The audit file is created in the same directory as the program, but the file is not populated yet.
Main Course	1. The election results have been processed (see UC_PARSE_IRV and UC_PARSE_CPL) 2. An audit file with the date and election type as its name is created (see EX1).
Alternate Courses	No alternate courses.
Exceptions	EX1: The program does not have write permissions.

	1. Print to the console that an audit log could not be created due to lacking permissions, then wait 5 seconds for the user to grant them. 2. Return user to Main Course step 2.

Name	Populate the audit file
ID	UC_POPULATE_AUDIT
Descriptions	Populate the audit file with relevant election information so that the election process can be verified and is clearer.
Actors	Election official
Organizational Benefit	Verifiability of the election and software.
Frequency of Use	Every time the software is used.
Triggers	The software is used.
Preconditions	The audit file has been created and opened in the program.
Postconditions	The file is populated with election information and how results were produced.
Main Course	1. The file is currently open in the program. (See CREATE_AUDIT) 2. The audit is filled with IRV election information, such as the election type, individual ballot information, and election-type specific aspects, such as rankings per candidate or how votes were transferred. (See AC1: and USE CASE 7) 3. The file is closed in the program. 4. The audit is now available to read.
Alternate Courses	AC1: The election type is CPL instead of IRV. 1. The audit is filled with CPL election information, such as the election type, individual ballot information, and election-type specific aspects, such as the party listing or how many seats a party gets. 2. Return user to Main Course 3.
Exceptions	No exceptions.

Name	Determine winner in tie breaker

ID	UC_TIE
Descriptions	In the case of a tie, handling the random coin flip to determine the winner.
Actors	Auditor, Election official, Tester, System Engineer
Organizational Benefit	Helps handle any ties in a fair manner that is more efficient and unbiased than if done by a person.
Frequency of Use	Whenever there is a tie that needs to be broken in an election.
Triggers	● CPL tie in allocating a seat ● CPL more seats than candidates and have to lottery it off ● Tie for lowest number of votes in round in IRV (assuming no candidate has a majority) ● Tie for winner in majority and popularity in IRV
Preconditions	There must be a tie between two or more parties that must be broken.
Postconditions	A winner is determined randomly.
Main Course	1. There is a tie somewhere in the system that must be broken 2. The system performs a random coin flip to determine the outcome of the tie 3. The system continues moving forward
Alternate Courses	AC1: There is a tie in allocation a seat in CPL 1. System uses a random coin flip to determine which party gets the seat 2. The system logs the results of who gets the seat 3. System logs the results of the rest of seats and vote counts 4. System outputs the results as well as vote counts to the terminal and audit file. AC2: CPL has more seats than candidates and have to lottery it off 1. System uses a random coin flip starting with the highest parties who didn't earn a seat to lottery off the excess seat or seats 2. The system logs the results of who gets the seat(s) 3. System logs the results of the rest of seats and vote counts 4. System outputs the results as well as vote counts to the terminal and audit file. AC3 Tie for lowest votes in round in IRV (when no candidate has majority) 1. In one of the rounds for the IRV, the candidates with the lowest votes are tied. No candidate has majority so the

	system uses the tiebreaker to determine which candidates should be eliminated and have their votes redistributed. 2. The loser is eliminated and their votes are reallocated. 3. The system logs the results of the coinflip and continues with the IRV election process. AC4 Tie for winner in majority and popularity in IRV 1. There is a tie between the final candidates in an IRV in both majority and popularity. 2. The system performs a random coin flip to determine the winner of the election. 3. The winner is logged, as well as the tie breaker and the system continues with the final processes of the IRV. AC5 More than 2 parties are tied in any given situation. 1. The main course is continually executed to determine a winner to the multi-way tie in a “tournament style” which means continually choosing pairs until only one remains. 2. The winner is logged and the program continues.
Exceptions	No exceptions.

Name	Determine the winner in IRV
ID	UC_IRV_WINNER
Descriptions	Determine the winner of an Instant Runoff Voting election
Actors	Auditor, Election Official
Organizational Benefit	Determine the winner for an IRV
Frequency of Use	Whenever there is an IRV election
Triggers	IRV election file is submitted to the program
Preconditions	IRV election file is submitted and was previously parsed to obtain election metadata and ballot data.
Postconditions	The winner for the IRV election is determined and output to the terminal and audit file. The number of votes and history is also logged.
Main Course	1. The number of first place votes for each candidate are compared and they are ranked accordingly 2. The candidate with the majority (>50%) of first place votes is determined as the winner

Alternate Courses	AC1: There is no clear majority but more than two candidates left 1. Candidate with the fewest counted votes is eliminated. 2. The voters who voted for the eliminated candidate have their next choice votes redistributed to the remaining candidates. 3. Repeat this process until a candidate has a majority (main course step 2) or two candidates are left and neither has a majority (AC2) or two candidates are left and they are tied for votes AC2: Two candidates remain but neither has a majority of votes. 1. Execute main course of use case UC_IRV_POPULARITY_VOTE. AC3 Two candidates remain but are tied for votes 1. The tie is broken via UC_TIE.
Exceptions	EX1: There is an unregistered candidate 1. Their votes are discarded and ignored

Name	Popularity vote if no majority in IRV
ID	UC_IRV_POPULARITY_VOTE
Descriptions	Determine the winner in IRV via popularity vote if there is no clear majority (>50% of votes) with only two remaining candidates
Actors	Auditor, Election Official
Organizational Benefit	Determine the winner for an IRV via popularity vote if no clear majority. This ensures that a fair result is always possible.
Frequency of Use	Whenever there is an IRV election and no clear majority
Triggers	No clear majority in IRV election
Preconditions	IRV election file is submitted and contents have been parsed. IRV election algorithm doesn't result in any clear majority for a candidate when two candidates remain.
Postconditions	The winner for the IRV election is determined and output to the terminal and audit file. The number of votes and history is also logged.
Main Course	1. The IRV election algorithm is run on the parsed data (see use case UC_IRV_WINNER) 2. There is no candidate with a clear majority (>50% of votes)

	when only two candidates remain. 3. Popularity of votes is used to determine the winner
Alternate Courses	AC1: There is a tie in popularity vote 1. Use the tie breaker (UC_TIE)
Exceptions	EX1: The program is closed mid popularity decision 1. The code must be rerun

Name	Determine winner(s) in CPL
ID	UC_CPL_WINNER
Descriptions	Determine the winner(s) for a Closed Party List election from the data in the CSV file.
Actors	Auditor, Election Official
Organizational Benefit	Determine with winner for a CPL Election
Frequency of Use	Anytime a CPL Election is submitted via CSV to the program
Triggers	CPL Election information is submitted via CSV
Preconditions	CPL election file is submitted and all election metadata and ballot data has been parsed and is present in the program's memory.
Postconditions	The winner for the CPL election is determined and output to the terminal and audit file. The number of votes and history is also logged.
Main Course	1. The number of votes for each party are compared using the largest remainder formula and they are assigned their number of seats according to each party. This may lead to some unallocated seats. 2. The parties divvy out their seats to their candidates in the order they have and those candidates are registered as being elected 3. Parties with the largest remainders of votes are given the remaining seats. 4. The winning parties' information as well as that of the other elected candidates including number of votes received and percentage are output to terminal and logged in the audit file. 5. The history of counting the votes up until the winner is also stored in the audit file.

Alternate Courses	AC1: There is a tie between parties for a seat 1. Use a random coin flip tiebreaker (see UC_TIE) AC2: A party wins more seats than they have candidates 1. All officials for that party are elected 2. Remaining seats are lotteried off (see AC2: of UC_TIE)
Exceptions	No exceptions.

Name	Not enough people for seats won in CPL
ID	UC_SEAT_LOTTERY_CPL
Descriptions	Determine how to lottery off excess seats if a party earns more seats than candidates in CPL.
Actors	Auditor, Election Official
Organizational Benefit	Assist in the distribution of seats for CPL elections without leaving leftovers
Frequency of Use	Anytime a party earns more seats than they have candidates in a CPL election
Triggers	A party earns more seats than candidates in CPL
Preconditions	CPL election file is submitted and the election metadata and ballot data has been processed via the CPL algorithm. The votes are counted up and a party earns enough votes that they earn more seats than they have candidates to fill.
Postconditions	The extra seats are fairly given out to other parties in the CPL election
Main Course	1. The number of votes for each party are compared using the largest remainder formula and they are assigned their number of seats accordingly to each party. 2. The parties divvy out their seats to their candidates in the order they have and those candidates are registered as being elected. 3. One party is given more seats than they have candidates for 4. Their excess seats are lotteried off using the tiebreaker use case (UC_TIE) to other random parties in the election who were runner ups to the leading party.
Alternate Courses	AC1: Two or more parties have excess seats 1. They are still lotteried off to runner up parties.

	2. Proceed with main course.
Exceptions	EX1: More seats than candidates 1. All officials elected. 2. Proceed with main course.

Name	Parse file for election type
ID	UC_PARSE_ET
Descriptions	Information must be extracted from input files and the correct voting algorithms to use must be determined.
Actors	System, election officials
Organizational Benefit	Verifiability of the election type to reduce possibility of errors or invalid results.
Frequency of Use	Once per election.
Triggers	The voting system is receiving input files having voting data. The user does not manually give the election type.
Preconditions	The input files need to have election type data and be in the formats that could be interpreted and read by the voting system (.csv). The file is the correct election file.
Postconditions	The program has the type of election stored in memory and will continue on to read the rest of the metadata based on this type.
Main Course	1. The system has an open and valid .csv election file 2. The system parses the first line for election type. 3. System extracts election type data. 4. System then moves on to extract further metadata based on the election type.
Alternate Courses	No alternate courses.
Exceptions	No exceptions.

Name	Parse file for IRV information

ID	UC_PARSE_IRV
Descriptions	The IRV information must be extracted from input files which will be used to conduct Instant-Runoff Voting elections.
Actors	System
Organizational Benefit	Ensures that the system stores and utilizes voting preferences accurately of every vote.
Frequency of Use	Once per IRV election
Triggers	The voting system is receiving input files having voting data. The user did not enter the metadata manually.
Preconditions	The input files need to have IRV data and be in the formats that could be interpreted and read by the voting system. The file offset is pointed to the second line of the election file.
Postconditions	The IRV information gets extracted and utilized to carry out Instant-runoff Voting elections.
Main Course	1. The system has the correct .csv file open 2. The system parses 3 lines for IRV information. 3. System extracts IRV metadata 4. System utilizes extracted data to carry out Instant-Runoff Voting elections.
Alternate Courses	No alternate courses.
Exceptions	No exceptions.

Name	Parse file for CPL information
ID	UC_PARSE_CPL
Descriptions	The CPL information must be extracted from input files, which is then used to conduct Closed Party List elections.
Actors	System

Organizational Benefit	It ensures that the system uses and stores every voter's party affiliations accurately.
Frequency of Use	Once per CPL election
Triggers	The voting system is receiving input files having voting data. The user did not enter the metadata manually.
Preconditions	The input files need to have CPL data and be in the formats that could be interpreted and read by the voting system. The file offset will be on the second line.
Postconditions	The CPL information gets extracted and utilized to carry out Cumulative Voting elections.
Main Course	1. The system has a valid .csv election file open 2. The system parses 4 lines for CPL information. 3. System extracts CPL metadata 4. System utilizes extracted data to carry out Cumulative Voting elections.
Alternate Courses	No alternate courses.
Exceptions	No exceptions.

Name	Parse file for ballot information
ID	UC_READ_BALLOTS
Description	The program will read in the information for each ballot in the election file and store it internally for use in the election algorithms.
Actors	Programmers, system testers, or election officials.
Organizational Benefits	This use case is critical to having accurate election results. Having the program be able to read ballot information quickly and effectively ensures that the results calculated later are accurate.
Frequency of Use	This will occur every time the program is run with an election file.
Triggers	The program reads the file header and moves on to the ballot lines.
Preconditions	The program has opened the election file. The program has internalized the election metadata (obtained either from the user or

Name	Parse file for ballot information
ID	UC_READ_BALLOTS
	the file's header). The type of the election is known. The file offset is pointed to the first ballot line.
Postconditions	The program now has all of the ballots and their data stored in some sort of data structure.
Main Course	1. The program reads in a line from the file (one ballot). 2. The ballot's data is placed into an object. 3. The ballot object goes into a data structure. 4. Steps 1-3 are repeated until the end of the file's contents is reached. 5. The file is closed
Alternate Courses	No alternate courses.
Exceptions	EX1: There is such a large number of ballots that the program runs out of memory. 1. The program will print a message explaining the error. 2. The program will abort.