

4.1 Bringing File Into System

Name:	Bring file into system
ID	UC_001
Description	<p>The system will receive a file from an external source. This file will be entered through a prompt or through the command line. Afterward, the software will determine if the file is valid,</p> <p>A valid file is a CSV file.</p>
Actors	Election officials, programmers, testers
Organizational Benefits	Ability to open a ballot file in order to process and conduct the election in an accurate and efficient manner.
Frequency of Use	<p>Run multiple times during the year at normal election times and special elections.</p> <p>Programmers and testers will use it more often during developmental phases or testing phases.</p> <p>In the case of errors, this use case will be used more than once during a session.</p>
Triggers	<ol style="list-style-type: none">1. The command read_file <filename> has been inputted through prompting of file.2. An input file has been provided as an input to launching software on the terminal.
Preconditions	The ballot file exists in the same directory as the software.
Postconditions	The file is ready to be processed

Main Course	<ol style="list-style-type: none"> 1. The system determines if the input file coming in exist (see EX1). 2. The system determines if the input file string coming in has a .csv file extension. If not, append the .csv to the string (see EX2). 3. System provides feedback to the terminal user that their file has been provided. 4. System has successfully opened the file if no errors has occurred. 5. The system will start to process the ballot file. See use case Processing Ballot.
Alternate Course	<p>AC1: Input file provided as part of launching the software.</p> <ol style="list-style-type: none"> 1. The system determines if the input file coming in exist (see EX1). 2. The system determines if the input file string coming in has a .csv file. If not, append the .csv to the string (see EX2). 3. Terminal provides feedback of what file name was entered. 4. System successfully opened the file. 5. System will start to process the ballot file. See use case Processing Ballot.
Exceptions	<p>EX1: File not found</p> <ol style="list-style-type: none"> 1. The system tells the user that the input file name cannot be found. <p>EX2: File is not a .csv</p> <ol style="list-style-type: none"> 1. If the file coming in has an extension and is not a .csv file, inform the user that the file is invalid.

4.2 Processing Ballot

Name:	Processing Ballot
ID	UC_002
Description	After a file has been opened, the ballot file will be process accordingly to the election voting system that has been determined. During the process, each candidate or party will be tally up. Once all the ballots have been processed, the information will be sent to the following election voting system that's conducted to determine a winner(s) and any additional features the voting election system has.
Actors	Programmers and testers.
Organizational Benefits	Allow an efficient processing of the ballots and ready format details for the actual voting system that takes place.
Frequency of Use	As frequent as voting system election that are occurring.
Triggers	A csv file has successfully been opened by the software. A redistribution of ballots need to occur in an instant run off.
Preconditions	Ballots file is open.
Postconditions	Ballots and tally marks have been processed and can be send into a pertaining voting system to conduct.
Main Course	<ol style="list-style-type: none">1. First line processed is "IR" and proceed (see EX1).2. Parse the next 3 lines to determine the number of candidates, the candidates separated by commas, and the number of ballots in the file.3. After the 4th line of the file, start processing each line as follows:4. Split the ballot line by "," and save the ranking of each ballot.5. Add the indicated first preference to each candidate.6. Repeat step 4-5 till all ballots has been process.7. Return the vote of each candidate to instant run off voting system to conduct the election.

Alternate Course	<p>AC1 Closed Party</p> <ol style="list-style-type: none"> 1. First line processed is “CPL” then proceed (see EX1). 2. Parse the next 2 lines to determine the number of parties, the parties separated by commas. 3. Parse the next x (number of parties) lines for each candidates in ranked order for the corresponding party. 4. Parse the next 2 lines for the number of seats and number of ballots. 5. Split ballots by “,” and add one vote to the selected party in each ballot. 6. Return the vote of each party to closed party voting system to conduct the election. <p>AC2 Redistribution of Ballots for IR</p> <ol style="list-style-type: none"> 1. The candidate that is eliminated will have their votes redistributed. 2. Add the indicated second/third/fourth/... preference accordingly of each ballot to each candidate. 3. Return the new vote of each candidate excluding the eliminated candidate to the instant run off voting system.
Exceptions	<p>EX1 Invalid voting system</p> <ol style="list-style-type: none"> 1. First line does not match any of the courses’ first line. Inform the user the opened file’s first line and sends an error. Remove the file as an opened file in the system.

4.3 Running the Instant Runoff Voting System

Name:	Instant Runoff Voting System
ID	UC 003
Description	The program has finished processing ballot and determined to process it as an instant run off election. The winner will be determined, vote distribution, and tie breakers will occur.
Actors	Election official, testers and programmers.
Organizational Benefits	To effectively process the ballots and conduct the instant run off voting system for efficient and ease.
Frequency of Use	Run multiple times during the year at normal election times and special elections. Will be run multiple times by testers and programmers.
Triggers	The ballot use case has finished processing the ballot file and is an instant run off ballot.
Preconditions	Ballot has successfully been processed.
Postconditions	The status of IR voting steps is done. The winner is determined.
Main Course	<ol style="list-style-type: none">1. The system determines if there is a majority candidate, a candidate with 51% votes. If there is, the winner has been decided (EX1).2. If there is no majority, the candidate with the lowest votes gets eliminated and the ballots for the lowest voting candidate gets redistributed (See Process Ballot use case).3. If there is a tie between the lowest voting candidate, tie break (see use case breaking a tie). The person who won the coin toss will be eliminated.4. Repeat step 1-3 till a candidate has been decided.

	<ol style="list-style-type: none"> 5. If there is only 2 candidate and there is not a majority, do the popularity (see the popularity for IR use case). 6. Once there is a winner, display the results (see results screen use case). 7. Produce an audit file (see producing an audit file use case).
Alternate Course	There is no alternative course for the instant run off.
Exceptions	<p>EX1 System gets a percentage out of range (0,100%)</p> <ol style="list-style-type: none"> 1. System notified user that an error has occurred

4.4 Popularity Case for IR

Name:	Popularity case for IR
ID	UC_004
Description	If there is not clear majority in IR between only two candidates, then popularity will win between the two.
Actors	Election officials, programmers, testers
Organizational Benefits	Resolves the no majority case in the Instant Runoff Voting method.
Frequency of Use	As frequent as the case when there is not a majority in instant run off. This case rarely happens.
Triggers	No majority win between two candidates in instant runoff.
Preconditions	Two candidates that does not have a majority.
Postconditions	The winner is given by popularity.
Main Course	<ol style="list-style-type: none">1. There's not a clear majority in the IR to determine the winner of the election2. System decides about the winner based of the popularity percentage (EX1).3. In the case there's a tie between popularity, tiebreaker. The winner will win the election. (see determine the tie use case).4. Resolve the instant runoff election and indicate who was the winner along with their voting percentages.5. Show the winner in the audit file.
Alternate Course	There is no alternate course.
Exceptions	EX1 System gets a percentage out of range (0,100%) <ol style="list-style-type: none">1. System notified user that an error has occurred

4.5 Running the Closed Party Voting System

Name:	Close Party Voting System
ID	UC_005
Description	The program has finished processing ballot and determined to process it as a closed party voting system. The winning candidates, seat distribution for parties, tie breakers, and lottery will occur.
Actors	Election officials, testers, and programmers.
Organizational Benefits	Effectively conduct a process for the close party voting system in a correct manner.
Frequency of Use	<p>If no errors have occurred, it will be run multiple times during the year at normal election times and special elections. This is as often as the close party voting system is indicated.</p> <p>It will be run multiple times by testers and programmers.</p>
Triggers	The ballot use case has finished processing the ballot file and is a closed party ballot system.
Preconditions	Ballot has successfully been processed
Postconditions	The status of IR is done with seat allocation distributed to candidates among majority parties.
Main Course	<ol style="list-style-type: none">1. The system gathered ballots percentage for each party (see the processing ballots use case).2. Distribute seats to party based on the percentages of parties (EX1).3. In the case that there is an odd seat that is distributed to two party tied for the seat, tiebreaker (see the use case breaking a tie).

	<ol style="list-style-type: none"> 4. Assign the winning candidates from the assigned party to the seat from the order of candidates as listed in the ballot. 5. In the case there is a seat overflow, do a lottery for the seats (see the use case CPL Lottery). 6. Once all winner has been determined for the seats, display the results (see results screen use case). 7. Produce an audit file (see producing an audit file use case)
Alternate Course	There is no alternative course for the closed party listing.
Exceptions	<p>EX1 System gets a percentage out of range (0,100%)</p> <ol style="list-style-type: none"> 1. System notified user that an error has occurred

4.6 CPL Seat Allocation Overflow

Name:	Seat Allocation Overflow in Closed Party List
ID	UC_006
Description	In closed party listing, when there is an overflow of seats allocated to a party where all their candidates have already filled their seats, the system will conduct a lottery of potential parties in parliament for the open seats. Each winner of the lottery will receive one of the seats until all seats has been filled.
Actors	Election official, testers, and programmers.
Organizational Benefits	Ensures that all seats are filled in an unbiased manner.
Frequency of Use	As frequent as when the case of overflow seats for majority party.
Triggers	There are more allocated seats than all candidates given by the party.
Preconditions	There are open seats for the lottery.
Postconditions	All seats are filled.
Main Course	<ol style="list-style-type: none">1. The user receives a notification of seat allocation for overflow and that a lottery is conducted for the open seats.2. The system will run a lottery for each seat.3. For each seat, run a random picking 1000 times and record the results. On the 1001 time, the random party picked will receive the seat (AC1, EX1).4. Once all seats have been filled, the seat allocation overflow has been resolved.
Alternate Course	AC1 The winning party has all their candidate filled in a seat <ol style="list-style-type: none">1. Conduct step 3 of main course but remove the party with all allocated members from the lottery.

<p>Exceptions</p>	<p>EX1: The party that won the lottery does not exist in the list of party provided</p> <ol style="list-style-type: none"> 1. System notifies user that the party that won the lottery does not exist in the CPL <p>EX2: Result of lottery is biased</p> <ol style="list-style-type: none"> 1. If one party wins most of the time out of the 1000 recorded results, redo step 3 of the related course (See EX3). <p>EX3: Repeated biased lottery</p> <p>If the lottery is biased 5 times, give the seat and inform to the terminal there's a possibility of bias and display the 1000 results.</p>
-------------------	--

4.7 Determine The Tie

Name:	Determining the Tie
ID	UC_007
Description	Whenever there is a tie situation that has occurred between 2 things, i.e., a candidate or party, a fair coin will be tossed and determine who's the winner.
Actors	Programmers and testers.
Organizational Benefits	Determining a tie that is both efficient, unbiased, and fair that supports the election integrity.
Frequency of Use	Whenever there is a tie that has occurred during the software voting systems. Ties are rare so they are less frequent. Fair coin toss may be checked frequently by programmers and testers.
Triggers	In IR: <ol style="list-style-type: none">1. Instant Runoff is in popularity votes and both candidates have the same number of votes.2. Majority has not been determined and two candidates have the same number of lowest votes. In CPL: <ol style="list-style-type: none">1. Party with the same number of percentages with a contested odd seat.
Preconditions	There is a tie that needs to be broken.
Postconditions	A tie has been broken and the winner is of the tie is given.
Main Course	<ol style="list-style-type: none">1. Instant runoff is in popularity vote and both candidates have the same number of votes, assign heads to the first candidate, assign tails to the second candidate.2. Flip a coin 1000 times and record the results. Then flip a coin on the 1001 to determine who's the winner of the popularity vote (see EX1).3. Return the winner of the popularity vote.

Alternate Course	<p>AC1 IR Lowest Candidate Elimination Tied</p> <ol style="list-style-type: none"> 1. Assign head to the first incoming lowest candidate, then tails to the second incoming candidate. 2. Flip a coin 1000 times and then flip a coin on the 1001 to determine who's the winner of the coin flip (see EX1). 3. Return the winner of coin flip. <p>AC2 CPL Tied Party Seat</p> <ol style="list-style-type: none"> 1. Assign heads to the first incoming party and tails to the second incoming party. 2. Flip a coin 1000 times and record the results. Then flip a coin on the 1001 to determine the winner of the seat (see EX1). 3. Return the winner.
Exceptions	<p>EX1 Result of Coin Flip is Biased</p> <ol style="list-style-type: none"> 1. If one result is skewed, winning more than 60% of the time, redo step 2 of the related course (See EX2). <p>EX2 Repeated Biased Coin Flip</p> <ol style="list-style-type: none"> 1. If the coin flip is biased 5 times, record the result, and inform to the terminal there is a possibility of bias and display the coin flip results.

4.8 Display Results

Name:	Displaying the results
ID	UC_008
Description	<p>After a successful voting system has been conducted, the terminal will display the results of the election to the user.</p> <p>If there was a tiebreaker or popularity vote, the terminal will indicate as such and the outcome.</p>
Actors	Election officials, programmers, testers
Organizational Benefits	Effectively display the results in a clear and concise manner.
Frequency of Use	<p>Runs multiple times during the year at normal election times and special elections.</p> <p>Programmers and testers will use it more often.</p>
Triggers	<ol style="list-style-type: none">1. A voting system has successfully been conducted and the final results has been determined2. The cmd report_election has been input to the prompt
Preconditions	The voting system has declared the winner(s).
Postconditions	Screen displayed the election results with the winner(s).
Main Course	<ol style="list-style-type: none">1. In the case of instant runoff, display on the terminal all candidates with their ballots counts.2. Highlight the majority who had won the election.3. In the case of a tie, show the elimination of the lowest candidate.
Alternate Course	<p>AC1: Closed Party</p> <ol style="list-style-type: none">1. In the case of a closed-party voting system, display on the terminal all

	<p>parties with their ballots distribution.</p> <ol style="list-style-type: none"> 2. Display the distribution of seats to parties. 3. Display which candidate from the party got an allocated seat. 4. In the case of a tie, show the tie results and the aftermath. 5. In the case that there are more seats than party members, show the lottery results which party got which seats. <p>AC2: CMD report_election</p> <ol style="list-style-type: none"> 1. Determine if there's currently an open file (see EX1). 2. If the voting election was an instant run off, go to the Main Course. 3. If the voting election was a Closed Party, go to AC1.
Exceptions	<p>EX1: No opened file.</p> <ol style="list-style-type: none"> 1. The system tells the user that there is no currently opened file.

4.9 Producing an Audit File

Name:	Producing an Audit File
ID	UC_009
Description	To procure information such as the election's result, distribution of ballots, and any recordings of certain voting systems.
Actors	Election officials, programmers, testers
Organizational Benefits	Help to organize the result of the election and in preparation to produce a recorded file showcasing the detailed steps of the software for record-keeping and legitimacy reasons.
Frequency of Use	<p>Audit file will produce multiple times during the year at normal election times and special elections.</p> <p>Programmers and testers will use it more often during developmental phases or testing phases.</p>
Triggers	<p>1. A voting system has successfully been conducted and the results has been determined.</p> <p>2. The cmd write_election has been input to the prompt.</p>
Preconditions	The voting system has processed a voting system and has declared the winner(s).
Postconditions	An audit file is ready to be outputted and label.
Main Course	<ol style="list-style-type: none">1. If the voting system is an instant runoff, record the distribution of ballots to candidates.2. If there's no majority, show the lowest candidate who got eliminated and the redistribution of their ballots.3. In the case there is a tie, record the tie results and show who got eliminated.4. If a popularity vote was conducted, indicate popularity vote was used.

	<ol style="list-style-type: none"> 5. Indicate which candidate won the election. 6. Go onto labeling the audit file case.
Alternate Course	<p>AC1: Closed Party Voting</p> <ol style="list-style-type: none"> 1. If the voting system is Closed party, record the distribution of ballots to the party. 2. Show the percentage rate of the ballots, and the calculation of seat distribution. 3. In case there is a tie, record the tie results and show who got eliminated. 4. In case there is a seat allocation overflow, record the lottery results. 5. Indicate which candidate and party won the seats. 6. Go onto labeling the audit file case. <p>AC2: cmd write_election</p> <ol style="list-style-type: none"> 1. The software determines if a file has been opened (see EX3). 2. Depending on the election type, follow main course or AC1.
Exceptions	<p>EX1 System fails on recognizing the type of election.</p> <ol style="list-style-type: none"> 1. The system notifies the user that an error has occurred. <p>EX2 System fails to produce the audit file.</p> <ol style="list-style-type: none"> 1. The system notifies the user that an error has occurred. <p>EX3 System has no opened file.</p> <ol style="list-style-type: none"> 1. The system notifies the user that there is no current opened file for an audit file to be produced. This error occurs if the user does write_election without an inputted file.

4.10 Labeling Audit File

Name:	Labeling Audit File
ID	UC_010
Description	Label a processed audit file by the type of voting election that had occurred and the date it was conducted. The audit file will then be output into the same directory as the csv file in readable permissions.
Actors	Election officials, programmers, testers
Organizational Benefits	Output a readable only file that holds the detailed documentation of what an audit file had procure. It will help organize results of the election and record-keeping for legitimacy reasons.
Frequency of Use	It will run as often as each time an election result has occurred, and the production of an audit file through normal means or through the command write_election
Triggers	An audit file has been produced and recorded the detailed steps for an election.
Preconditions	The audit file is processed/produced.
Postconditions	The audit file is correctly labeled in the same directory as the .csv file.
Main Course	<ol style="list-style-type: none">1. In the case that the audit file produced was for instant runoff, start the label with “IR”.2. Get the meta data of the .csv file that was inputted and obtain the date (See EX1).3. Append the date to the label of the audit file.4. Output the audit file with its label, i.e., “IR02032023” in the same directory as the .csv file with permission readable only.

Alternate Course	<ol style="list-style-type: none">1. In the case that the audit file produced was for closed party listing, start the label with “CPL”.2. Get the meta data of the .csv file that was inputted and obtain the date.3. Append the date to the label of the audit file.4. Output the audit file with its label, i.e., “CPL02032023” in the same directory as the .csv file with permission readable only.
Exceptions	<p>EX1 CSV meta data unable to be accessed</p> <ol style="list-style-type: none">1. Inform the user that the csv file date was unable to be accessed. Give the date of today for the audit file (See EX1).