Use Cases Document for Voting System

Prepared by Jacob Malin, Romeo Sweeney,
Nathan Kjaer, Shruti Jayavardhanan

Name	Command Line File Input
ID	UC_INPUT_CLI
Description	A method for testers to use for inputting a file that works from the command line.
Actors	Testers, Programmers
Organizational Benefits	Allows testers and programmers to test the system using a command line prompt. Allows testers to run automated scripts to test different parts of the system.
Frequency of Use	As often as the system is tested using scripts or a command line argument is given manually by a tester.
Triggers	An actor runs the program from the command line.
Preconditions	None
Postconditions	The file will be verified to exist and be a valid election file.
Main Course	1. Ensure that there is 1 command line argument (see EX1). 2. Extract the file name from the command line argument array. 3. Verify that the file exists within the correct directory (see EX2).
Alternate Courses	None
Exceptions	EX1 There are more than one command line arguments. 1. Inform the user that there is an incorrect number of command line arguments. 2. Resume at UC_INPUT_TEXT Main Course. EX2 The file does not exist in the correct directory. 1. Inform the user that the file was not able to be found. 2. Resume at UC_INPUT_TEXT Main Course.

Name	Text Prompt File Input
ID	UC_INPUT_TEXT
Description	A method for users to use when inputting a file that works from inside the program.
Actors	Election Officials
Organizational Benefits	Allows Election Officials to input a file into the program in an organized method.
Frequency of Use	Whenever an Election official wants to run an election file.
Triggers	An election official runs the program, no command line argument is given, or the use case UC_INPUT_CLI or UC_EXTRACT follows an exception to this use case.
Preconditions	None
Postconditions	The file will be verified to exist and be a valid election file.
Main Course	1. The system will prompt the user to input a text file. 2. The system will extract the file name from the user's input. 3. The system will verify the file exists within the correct directory (see EX1).
Alternate Courses	None
Exceptions	EX1 The file does not exist in the correct directory. 1. Inform the user that the file could not be found. 2. Prompt the user again. 3. Resume at Main Course Step 2.

Name	Extract Info
ID	UC_EXTRACT
Description	Extract info from a given file and store for later processing.
Actors	System
Organizational Benefits	File is read and closed before processing the election, which allows for fewer opened files at once, and better separation of duties. All information from the file is stored in a table, which is easier to access than from the file.
Frequency of Use	Once every time an election is run.
Triggers	The file name is read in from UC_INPUT_CLI or UC_INPUT_TEXT.
Preconditions	File name is known and the file exists in the current directory. The file is a valid election file.
Postconditions	Information from the file will be stored. The votes will be tallied for parties and candidates. The file is closed.
Main Course	1. Open the file for reading (see EX1). 2. Dump file contents (see EX2). 3. Close the file (see EX3). 4. Store CPL/OPL. 5. Store the number of seats, ballots, and parties (see AC1). 6. Store party names/candidates. 7. Tally and store the votes for parties, candidates, and the total number of votes.
Alternate Courses	AC1 This is an OPL election. 1. Store the number of seats, ballots, and candidates. 2. Resume at Main Course Step 6.
Exceptions	EX1 The file open errors. 1. Close the file. 2. Inform the user that the file was not able to open. 3. Resume at UC_INPUT_TEXT Main Course. EX3 The file read errors. 1. Close the file. 2. Inform the user that the file was not able to be read. 3. Resume at UC_INPUT_TEXT Main Course. EX3 The file close errors. 4. Inform the user that the file was not able to close. 5. Resume at UC_INPUT_TEXT Main Course.

Name	CPL
ID	UC_CPL
Description	A program to evaluate an election based on Closed Party Listing rules.
Actors	System
Organizational Benefits	Turns the ballot list into results of the election, making the winners of the election clear.
Frequency of Use	Whenever a file is in CPL format, therefore whenever a CPL election is run.
Triggers	The use case UC_EXTRACT completes, and the election type is CPL.
Preconditions	Information from the file is stored.
Postconditions	The CPL results have been calculated and the winners are known.
Main Course	1. Divide the number of votes by the number of seats to obtain the quota (see EX1). 2. Divide each party's votes by the quota to obtain both the first allocation of seats and the remainder (see AC1). 3. Calculate the remaining number of seats. 4. Allocate the remaining seats based on the remaining votes in decreasing order (see AC2, AC3, EX2). 5. Add up the final seat totals. 6. Distribute the seats in each party in the order listed in the input file.
Alternate Courses	AC1 There are more seats in the first allocation than party members or an equal number of seats and party members. 1. The number of seats allocated is set to the number of party members. 2. The party is removed from consideration for further seats. 3. Resume at Main Course Step 3. AC2 There are no remaining seats. 1. Resume at Main Course Step 6. AC3 There is a tie. 1. Break the tie: refer to UC_COIN_FLIP. 2. Then resume at Main Course Step 5.
Exceptions	EX1 There are no votes or no seats. 1. All parties get 0 seats. EX2 All parties have been removed from consideration for further votes. 1. The remaining seats are not allocated. 2. Resume at Main Course Step 6.

Name	OPL
ID	UC_OPL
Description	A program to evaluate an election based on Open Party Listing rules.
Actors	System
Organizational Benefits	Turns the ballot list into results of the election, making the winners of the election clear.
Frequency of Use	Whenever a file is in OPL format, therefore whenever an OPL election is run.
Triggers	The use case UC_EXTRACT completes, and the election type is OPL.
Preconditions	Information from the file must be stored.
Postconditions	The OPL results have been calculated and the winners are known.
Main Course	1. Divide the number of votes by the number of seats to obtain the quota (see EX1). 2. Divide each party's votes by the quota to obtain both the first allocation of seats and the remainder (see AC1). 3. Calculate the remaining number of seats. 4. Allocate the remaining seats based on the remaining votes in decreasing order (see AC2, AC3, EX2). 5. Add up the final seat totals. 6. Order the candidates by popularity in each party (see AC4). 7. Distribute the seats in each party in the order calculated.
Alternate Courses	AC1 There are more seats in the first allocation than party members or an equal number of seats and party members. The number of seats allocated is set to the number of party members. The party is removed from consideration for further seats. Resume at Main Course Step 3. AC2 There are no remaining seats. 1. Resume at Main Course Step 6. AC3 There is a tie. 3. Break the tie: refer to UC_COIN_FLIP. 4. Then resume at Main Course Step 5. AC4 There is a tie. 1. Break the tie: refer to UC_COIN_FLIP. Then resume at Main Course Step 7.
Exceptions	EX1 The file is for a CPL election. 1. Resume at step 4 of UC_CPL. EX2 The OPL election fails to run. 1. The system sends out a notification about the failure. 2. Returns to Main Course 1

	EX3 Election results are not written. 1. The system sends out a notification about the failure. 2. Return to Main Course step 2.

Name	Produce Audit File
ID	UC_AUDIT
Description	The system generates an audit file containing the election details (such as seat allocations, winners, and relevant statistics).
Actors	System
Organizational Benefits	Provides transparent records of the election process and can be used to communicate with media people without giving private info.
Frequency of Use	At the end of each election process.
Triggers	Completion of seat allocation and winner decision process, either use case UC_OPL or UC_CPL.
Preconditions	Winners and seat allocations have already been determined.
Post Conditions	An Audit File is generated and saved with all the information outline in the Main Course. This file is assumed to be correct and contains all relevant information.
Main Course	1. Open an audit file for writing (see EX1). 2. Write the type of election. 3. Write the number of parties, ballots, and seats. 4. Write the candidates' names and party affiliations. 5. Write the calculation for the largest remainder approach. 6. Write a list of the seat winners and party affiliations (see AC1). 7. Save the file (see EX2).
Alternate Courses	AC1 This is for OPL. 1. Also write the number of votes that each candidate received. 2. Resume at Main Course Step 7.
Exceptions	EX1 File does not open. 1. Inform the user that the audit file was not able to be opened. 2. Resume at UC_INPUT_TEXT Main Course. EX2 File isn't saved properly. 1. Inform the user that the audit file was not able to be saved. 2. Resume at UC_INPUT_TEXT Main Course.

Name	Display results
ID	UC_DISPLAY
Description	The system displays election results to the screen (such as seat allocations, winners, and relevant statistics).
Actors	System
Organizational Benefits	The user can view the results in a readable format.
Frequency of Use	At the end of each election process.
Triggers	Completion of seat allocation and winner decision process, either use case UC_OPL or UC_CPL.
Preconditions	Winners and seat allocations have already been determined.
Post Conditions	The results are displayed on the screen.
Main Course	1. Display the type of election and number of seats. 2. Display the number of ballots cast and the winners. 3. Calculate stats for all candidates and display them to the screen. This includes number of votes, percentage of votes, and if they won.
Alternate Courses	None
Exceptions	None

Name	Coin Flip (Tie)
ID	UC_COIN_FLIP
Description	The system provides a mechanism for handling ties via a "coin flip"
Actors	System
Organizational Benefits	Allows for ties in a CPL or OPL election.
Frequency of Use	Each time a tie occurs.
Triggers	Use cases UC_OPL or UC_CPL encounter a tie when two or more candidates have the same number of votes cast. Use cases UC_OPL or UC_CPL encounter a tie when two or more parties have the same number of remaining votes.
Preconditions	OPL or CPL must be actively running.
Post Conditions	A tie is handled fairly and the chosen candidate receives the seat.
Main Course	1. A tie is encountered where one or more candidates receive the same number of votes. 2. Compute a random number for each candidate after looping 1000 times to ensure fairness (see EX1). 3. Compute a random number that determines the winner after looping 1000 times to ensure fairness (see EX1, EX2). 4. The winner is the one closest to the system-generated number. 5. Assign the winner their seat. 6. Resume as described in the respective use case.
Alternate Courses	None
Exceptions	EX1 The system fails to loop 1000 times to ensure fairness. 1. Return to Main Course Step 2. EX2 The random number generation ties. 1. Return to Main Course Step 2.