| Name | Run Open Party Listing Voting |
| --- | --- |
| ID | **UseCase\_Team9\_A1** |
| Description | * This is where citizens vote for their preferred candidate of their choice. The candidates may belong to a specific party or independent. |
| Actor(s) | * Election official |
| Preconditions | * The CVS file specifies type of voting. I.e. OPL * Other condition are met for the file to be processed :   + Type of Vote   + Number of candidates   + Name of the candidate, part name ( Initial )   + Number of seats available   + Number of ballots * The above condition should be separated by newline * Assumption : No error in the ballots   + Such that, Only a single 1 will be placed in the position of the given candidate |
| Basic flow | * accepts CSV file as an input and checks if all requirements. ( Requirement is listed in pre-condition) * Program will identify candidates and count individual vote plus party vote * Also, the system will calculate the number of seats available for each party.   + Identifying number of seats per each party     - Largest reminder formula ( Quota )       * quote = total number of votes in a district / total number of seats         + 1st allocation will be Floor(quote) ->         + 2nd allocation -> distribute remaining seats to the party that has largest reminder * An audit and report file are generated. Both files will be for READ ONLY * System will display who won among the candidates |
| Alternate flow | * Case 1 : Candidates with the highest votes wins * Case 2 : Tie ?   + Refer to UseCase\_Team9\_L2 |
| Exception flows | * Invalid input file -> prompt the user to select or enter file name |
| Postconditions | * We will have a winner after processing the input file. * There will be an audit and report files generated respectively to show calculation was done   + These files will be available for read only. |

| **Name** | Input CSV File |
| --- | --- |
| **ID** | **UseCase\_Team9\_I1** |
| **Description** | Allows user to use a CSV file which contains the election votes to determine the winner of an election. This will also be able to |
| **Actor(s)** | Election Officials |
| **Organization**  **Benefits** | Will allow a user to input a CSV file into the program. The program will read the type of election and run the chosen election type. From the CSV file, the program will also read in the votes for each candidate so that it can determine the winner of the election. |
| **Frequency of Use** | Will only be used once per election run. |
| **Trigger** | The user will type in the program name along with the CSV file so that it can be entered. |
| **Precondition** | CSV file must be in the same directory as the program executable. CSV file must also have the correct format with no mistakes. |
| **Postcondition** | Program will read in the CSV file and the election votes will be organized and counted. |
| **Main Course** | 1. User enters program executable along with the CSV file name in the terminal (EX1) |
| **Alternate Course** | N/A |
| **Exceptions** | EX1 User enters an invalid file   1. Program tells user that file is invalid or not found. 2. Program does not run and the election votes are not counted. 3. Allows user to re-run the program executable |

| **Name** | Return results to display |
| --- | --- |
| **ID** | **UseCase\_Team9\_I2** |
| **Description** | Once a winner is determined by the program, it will need to display the results to the screen/terminal of where the program was run. These results may include the rankings of the election, their names and party, type of election, and number of ballots. |
| **Actor(s)** | System |
| **Organization**  **Benefits** | Immediately show results of the election to the user instead of having to wait for the media or open the audit file. |
| **Frequency of Use** | Once after the program completes counting the votes and an election winner is chosen. |
| **Trigger** | Once all the votes are counted and based on the voting style, the program will determine the winner of the election. Once the winner is chosen and an audit file is created, the results of the display should show on the screen. |
| **Precondition** | CSV file is read in the program, the votes have been counted, there is a conclusion for the election winner, and an audit file has been created. |
| **Postcondition** | The results will be displayed on the screen. |
| **Main Course** | 1. Program reads in CSV file 2. Depending on OPL or IRV, the election will be run 3. Winner will be chosen 4. Audit file is created 5. All results will be displayed to the screen |
| **Alternate Course** | N/A |
| **Exceptions** | N/A |

| Name | Run Instant Runoff Voting |
| --- | --- |
| ID | **UseCase\_Team9\_L1** |
| Description | Instant runoff voting will be conducted with the file of votes provided. |
| Actor(s) | Election Official |
| Preconditions | The election official possesses the voting results in CSV format with 1st Line: specifying “IR.” 2nd line: Number of Candidates. 3rd Line: The candidates separated by commas. 4th Line: Number of ballots in the file. We can assume there are no errors in the ballots. Each ballot will have at least 1 ranking. The ranking numbers will not have issues. |
| Basic flow | 1). System will parse the CSV and create identifiable Voters complete with choice(s) of candidates  2). System will count and distribute ballots for each candidate starting with the primary choice.  3). All results are logged for an audit report.  4). The least popular candidate will be marked as eliminated and their ballots are distributed among the other candidates based on the next highest vote level.  5). The system logs final results and then finishes if there is only one candidate left. Otherwise, return to step 4.  6). System will display to the screen the results of the election and output an audit log. |
| Alternate flow | Case 1: Tie  1). At step 5 of the basic flow, there may be a tie. Ties will be resolved using a tiebreaker (see UseCase\_Team9\_L2). The eliminated candidate’s ballots will be distributed among valid candidates  2). If no tie breaker can be found, the winning candidate is decided by random draw. The alternate flow now rejoins the basic flow at step 4  Case 2: Tester or Developer wants to Shuffle or Manual Test  1). At step 3 of the basic flow, the parsed voter profiles will be shuffled (see UseCase\_Team9\_L3) |
| Exception flows | Invalid information is entered  1). System cancels vote process  2). System alerts user to the to the field that needs attention  3). User is given the option of re-entering information |
| Postconditions | Results are generated for viewing and data is kept for auditing purposes. Produce an audit file with the election information at the time (e.g. Type of Voting, Number of Candidates, Candidates, Number of Ballots, calculations, how many votes a candidate had, etc), you should list the winner(s), and show how the election progressed so that the audit could replicate the election itself. You ballot information and its order of being received if applicable. |

| Name | Coin Flip Tie Breaker |
| --- | --- |
| ID | **UseCase\_Team9\_L2** |
| Description | A tie breaker is decided among two or more candidates |
| Actor(s) | System |
| Preconditions | Two or more candidates are submitted for a tie breaker |
| Basic flow | 1). System records the list of candidates in a tie  2). If there is only one candidate in the list, then they are returned as the winner. Skip to step 5  3). 1001 coins are flipped for the first two candidates in the list. If the number of heads is greater than tails then the first candidate wins, and if the number of tails is greater than heads the second candidate wins.  4). The winner is added back to the end of the list and the loser is eliminated. The update is logged and recorded for audit. Return to step 2  5). Output the winner back to the caller |
| Alternate flow | No alternate flows considered at this time. |
| Exception flows | No exception flows considered at this time. |
| Postconditions | Given that at least two candidates were input, there will be one winner as the output. The audit will reflect the actions taken. |

| Name | Shuffle Votes |
| --- | --- |
| ID | **UseCase\_Team9\_L3** |
| Description | The order in which voters are counted is shuffled |
| Actor(s) | Tester, Programmer |
| Preconditions | A list of valid voters and their associated votes post-parse exist |
| Basic flow | 1). System selects a new order for the voter list to be in  2). System ensures that it is different from the current  3). System returns the shuffled order |
| Alternate flow | Case 1: The shuffled order is the same as the unshuffled order  1). Shuffle again |
| Exception flows |  |
| Postconditions | A list of valid voters and their associated votes are returned in a shuffled order |

| Name | View Audit |
| --- | --- |
| ID | **UseCase\_Team9\_R1** |
| Description | The election officials view a detailed audit of the election results |
| Actor(s) | Election Officials |
| Organization Benefits | Election officials can view an audit of the election results to ensure everything went well. |
| Frequency of Use | Every time an election is run under the voting system software an audit will be created if desired. |
| Trigger | A program will run through the election results and a programmer will have an option to generate an audit that can be sent directly to the election officials. |
| Precondition | A vote count file has just been run and the election official says yes to the creating of the audit. |
| Postcondition | The election official will have read the audit file and then can exit the program. |
| Main Course | 1. Election official finishes running election counting result program  2. Election official then can create an audit file  3. The audit file will be stored locally as a txt file  4. The election official may then view the audit file (EX1)  5. The election officials will look over the audit to make sure that everything is accurate. |
| Alternate Courses | The election has already been run but an audit has not been run on it. The official will rerun the election with the vote count file.  1. The voting system program can be run as many times as the user desires. |
| Exceptions | EX1 Program crashes  1. The official will run the program again with the same file.  2. They will then return to step 3 of the main course. |
|  |  |

| Name | Create Audit |
| --- | --- |
| ID | **UseCase\_Team9\_R2** |
| Description | Create an audit for the voting system |
| Actor(s) | Election Officials |
| Organization Benefits | The process of the election will be documented fully so that officials can view the results to make sure they are valid. |
| Frequency of Use | An audit can be generated every time an election runs through the system. |
| Trigger | Officials can run the election and then will be prompted whether or not they want to run the audit. |
| Precondition | An election will have just been run. |
| Postcondition | An audit is now available as a text file and can be sent anywhere the official wishes. |
| Main Course | 1. Official has just finished running an election (EX1)  2. Official will then be prompted if they want to create an audit file or not.  3. Then the audit file will be created.  4. The official will then be prompted about whether or not they want to generate a media report and the program will then terminate. |
| Alternate Courses | The election has already been through the program, the programmer can then rerun the election and then generate an audit file.  1. Program can be run on the same file as many times as desired. |
| Exceptions | EX1 Voting Fails  1. An error occurs at some point so the official can just rerun the program.  2. If the program runs properly they can go back to step 2. |
|  |  |

| Name | Create Media Report |
| --- | --- |
| ID | **UseCase\_Team9\_R3** |
| Description | Create a report of the election results for the media. |
| Actor(s) | Election Officials |
| Organization Benefits | The media will get an easy snapshot of the election results as opposed to a long and complex audit report. |
| Frequency of Use | Every time an election runs through the system the election officials can create a media report. |
| Trigger | The official will have just finished running the election through the system and answering a prompt about generating an audit.  A prompt will appear if they want to generate a media report. If yes a text file will be generated and the report can be sent to the media that requested it. |
| Precondition | An election will have just been run and the official answered “yes” to the prompt about generating a media report. |
| Postcondition | A report is now available as a text file and can be sent anywhere the official wishes. |
| Main Course | 1. Official has just finished running an election  2. Official will then be prompted if they want to create an audit file or not.  3. Then the audit file will be created or not created.  4. The official will then be prompted about whether or not they want to generate a media report and the program will then terminate.  5. If yes a report will be generated that will be stored locally for the official to send out. |
| Alternate Courses | The election has already been through the program, the official can then rerun the election and then generate a report.  1. Program can be run as many times as the user desires. |
| Exceptions | EX1 Voting Fails  1. Official will be asked if they want to rerun the election.  2. If the election is rerun they will go back to step 4 of the main course. |
|  |  |