

Main Use Cases:

1. Users input required information for voting system to run election <Q - Ask if only this one or each item on screen is a use case.> - OK
  - a. Others: import ballots, number of seats, choose algorithm  
<Q - Do users manually load files by specifying file location and select files? Or does the system load files from a predefined location? Or does the system poll a predefined location?>
2. Users run SVT election - OK
3. Users run plurality election
4. Users run test file(s) - OK
5. User turn off shuffle ballots - OK
6. The voting system displays election results to users - OK
  - a. UC\_06: Provide elected / non-elected lists to user <Q - not sure on this one. Ask how they want the information do determine if a use case or not> should this be a separate case?
7. The voting system displays help window for users - OK
8. The voting system provides election audit file to users - OK
9. ??? The voting system clears inputs and results for another round of election/test - <Q - does it need to reset?>

Functional requirements:

The system should be able to:

- Run a plurality election type.
- Run an STV election type.
- Provide the ability to break ties.
- Calculate election statistics
- Accept user inputs
- Shuffle ballots

Non-functional requirements:

The system should operate given the following function and performance constraints:

- Accept CSV ballot files in Windows Comma Separated format.
- The election needs to be processed in under 5 minutes.
- Run on a CSE lab machine from the command prompt.
- Must process up to 100,000 ballots.

Use cases: (Josh)

- UC\_1: Input how many seats to fill/algorithm to use
  - UC\_2: Import ballots from files
- UC\_3: Request audit report
- UC\_4: Run test file "Calibrate system"
  - Toggle shuffle option
- UC\_5: Display election results
- UC\_6: Display help window
- 
- Run plurality voting ??
- Run Droop Quota voting ??
- Shuffle ballots ??

Roles / Users (Bryan):

- Election Official
- Developer / Tester

Use cases (Bryan):

- UC\_01: Display "Gather Information" screen
  - UC\_09: Import number of seats <Q - Can this be included in UC\_01?>
  - UC\_10: Import election type to run <Q - Can this be included in UC\_01?>
  - UC\_11: Import ballots <Q - Can this be included in UC\_01?>
- UC\_02: Run election
- UC\_03: Toggle ballot shuffle <Q - Which users want this? Command line option or menu?>
- UC\_04: Display election results
  - UC\_06: Provide elected / non-elected lists to user <Q - not sure on this one. Ask how they want the information do determine if a use case or not>
- UC\_05: Provide STV election audit
- UC\_06: Provide elected / non-elected lists to user <Q - not sure on this one. Ask how they want the information do determine if a use case or not>
- UC\_07: Display help screen
- UC\_08: Run test cases <For developers only>
  - UC\_03: Toggle ballot shuffle <Q - Which users want this? Command line option or menu?>
- UC\_09: Import number of seats <Q - Can this be included in UC\_01?>
- UC\_10: Import election type to run <Q - Can this be included in UC\_01?>
- UC\_11: Import ballots <Q - Can this be included in UC\_01?>

Use cases (Hailin):

1. User input information into system

2. System loads ballot files <Q - Do users manually load files by specifying file location and select files? Or does the system load files from a predefined location? Or does the system poll a predefined location?>
3. System process ballot files <Q - Should the system check if ballot files match election type?>
4. STV: System calculates droop quota
5. STV: System produces ballots for further processing (shuffling if debug option is off, no shuffling if debug option is on)
6. STV: System run Droop Quota election algorithm
7. STV: System generates an audit report.
8. STV: Break ties
9. STV: Command line input to turn on debugging mode
10. Plurality: System run plurality election algorithm
11. Plurality: Break ties
12. System calculates candidates stats
13. System display outputs
14. System display help window

Functional requirements (Hailin): <Q- How detailed do functional requirements need to be?>

1. System shall be able to read in csv file(s)
2. System shall be able to take user inputs
3. System shall be able to run STV voting
  - a. System shall shuffle ballots
  - b. System shall calculate Droop Quota
  - c. System shall run Droop Quota algorithm
  - d. System shall generate an audit report
  - e. System shall be able to break ties
4. System shall be able to run plurality voting
5. System shall be able to break ties in plurality voting scheme
6. Users shall be able to run a test file with shuffle option off
7. System shall display election stats
8. System shall have a help window

Use cases (Colin):

- 1) User inputs seats to fill and algorithm to use
- 2) User selects the ballot file (what is the max number of files?)
- 3) Run election
- 4) Disable shuffle / Enter debug mode
- 5) Display election results
- 6) Display help window
- 7) Clear election result/reset (Do we need this?)

Name	Input required voting information
ID	UC_01
Description	Upon starting the system, the user must input information to correctly process the election.
Actors	Election officials
Organizational Benefits	The system gets information from the user to process the ballots for the election in the appropriate way.
Frequency of Use	Each time an election is run.
Triggers	The election official starts the voting system.
Preconditions	The voting system program has been started.
Postconditions	<p>The system will have the following information which is required to run the election:</p> <ul style="list-style-type: none"> <li>- The number of seats to fill in the election.</li> <li>- The type of election to perform.</li> <li>- The location of the election ballots.</li> </ul>
Main Course	<ol style="list-style-type: none"> <li>1. The system prompts the user to input the number of seats to fill in the election.</li> <li>2. The user enters the number of seats to fill. (see EX_01, EX_02)</li> <li>3. The system prompts the user to input the type of election to run.</li> <li>4. The user enters the type of election to run. (see EX_03)</li> <li>5. The system prompts the user to input the location of the files containing the election ballots.</li> <li>6. The user enters the location of the files containing the election ballots. (see EX_04, EX_05)</li> </ol>
Alternate Courses	<p>AC_01: The user closes the voting system program at any time</p> <ol style="list-style-type: none"> <li>1. The user restarts the voting system.</li> <li>2. Return to main course step 1.</li> </ol>
Exceptions	<p>EX_01: The user enters something other than a number</p> <ol style="list-style-type: none"> <li>1. The system notifies the user that they must enter a number.</li> <li>2. Return to main course step 1.</li> </ol> <p>EX_02: The user enters zero or another number that is not possible</p> <ol style="list-style-type: none"> <li>1. The system notifies the user that they must enter a valid number.</li> </ol>

	<ol style="list-style-type: none"><li>2. Return to main course step 1.</li></ol> <p>EX_03: The user enters an election type that is not supported by the system.</p> <ol style="list-style-type: none"><li>1. The system notifies the user that they must enter a valid election type.</li><li>2. Return to main course step 3.</li></ol> <p>EX_04: The user enters a non valid file path.</p> <ol style="list-style-type: none"><li>1. The system notifies the user that they must enter a valid file path.</li><li>2. Return to main course step 5.</li></ol> <p>EX_05: The user enters a file type that is not supported.</p> <ol style="list-style-type: none"><li>1. The system notifies the user that they must enter a valid file type.</li><li>2. Return to main course step 5.</li></ol>
--	---

Name	User runs SVT Election
ID	UC_2
Description	The user has set up an election and now wishes to run a SVT election
Actors	Election Officials
Organizational Benefits	Simplifies/automates the process of running an SVT election
Frequency of Use	Once per election
Triggers	User selects option to run election
Preconditions	UC_1 needs to be complete(number of seats has been specified, election type has been selected as SVT, ballot file has been loaded)
Postconditions	Election has been run, results are displayed
Main Course	<ol style="list-style-type: none"><li>1. User selects option to run election</li><li>2. System runs election, displays results when complete</li></ol>
Alternate Courses	AC_1: User did not input all required information into the system <ol style="list-style-type: none"><li>1. System produces an error and prompts the user to input missing information</li><li>2. Go to main course step 1</li></ol>
Exceptions	EX_1: Issues with ballot file <ol style="list-style-type: none"><li>1. Alert user that it couldn't read ballot file</li></ol>

Name	
ID	UC_3
Description	
Actors	
Organizational Benefits	
Frequency of Use	
Triggers	
Preconditions	
Postconditions	
Main Course	1.
Alternate Courses	AC_1:
Exceptions	EX_1:

Name	Users run test file(s)
ID	UC_4
Description	The users run a test file and the shuffle option should be able to be turned off.
Actors	Election officials, developers, testers
Organizational Benefits	Running test files allows users to ensure the system is calibrated properly and also allows for unit testing.
Frequency of Use	Not frequent. During system development and deployment. When the system is tested as needed.
Triggers	Users run command line argument to test system
Preconditions	The system can run elections. The ballot shuffling function can be turned off.
Postconditions	The system produces election results.
Main Course	<ol style="list-style-type: none"> <li>1. User starts the system</li> <li>2. User runs command line argument</li> <li>3. System loads test file(s)</li> <li>4. System processes test file(s) with options provided by user</li> <li>5. System produces election results</li> </ol>
Alternate Courses	<p>AC_1: User did not start the system before running command line argument</p> <ol style="list-style-type: none"> <li>1. Operating system does not recognize command</li> <li>2. Go to main course step 1</li> </ol> <p>AC_2: User run wrong command line argument</p> <ol style="list-style-type: none"> <li>1. System produces error</li> <li>2. Go to main course step 2</li> </ol> <p>AC_3: User did not provide enough information</p> <ol style="list-style-type: none"> <li>1. System produces error</li> <li>2. Go to main course step 2</li> </ol>
Exceptions	EX_1: User provided wrong options



Name	Turn off ballot shuffle
ID	UC_05
Description	The user decides to turn off the ballot shuffling feature of the voting system.
Actors	Election officials, developers, testers
Organizational Benefits	Turning off the ballot shuffle will create a scenario where the system will provide repeatable election results each time an election is ran with the same ballot information. This can be used for system calibration and testing.
Frequency of Use	Whenever system testing or calibration is needed.
Triggers	<p>&lt;Q - Can this use case be included in UC_01 as another piece of information to input into the system or can it be a command line option?&gt;</p> <p>The user starts the voting system using the appropriate command line option.</p>
Preconditions	The user decides that the ballot shuffle should be turned off. The appropriate command is entered into the system.
Postconditions	The ballots for an election will be processed in the order they have been entered into the ballot files.
Main Course	<ol style="list-style-type: none"> <li>1. The user enters the appropriate command into the system to turn off ballot shuffling.</li> <li>2. The system prompts the user to confirm they would like to turn off ballot shuffling.</li> <li>3. The user confirms that they would like to process the election without ballot shuffling. (see EX_01)</li> <li>4. The system turns off the ballot shuffling option.</li> </ol>
Alternate Courses	<p>AC_1: After starting the system and prior to running the election, the user decides to turn off the ballot shuffling option.</p> <ol style="list-style-type: none"> <li>1. The user must restart the voting system.</li> <li>2. Return to the main course step 1.</li> </ol>
Exceptions	<p>EX_01: The user inputs that they would not like to turn off ballot shuffling.</p> <ol style="list-style-type: none"> <li>1. The user must restart the voting system.</li> <li>2. Return to the main course step 1.</li> </ol>

Name	The voting system displays election results to users
ID	UC_6
Description	After an election is run the system will display the results of the election. It will display the type of election(STV/plurality), the number of ballots, the number of seats, the number of candidates and the winners.
Actors	Election Officials
Organizational Benefits	Allows election official to easily view the results of an election
Frequency of Use	Once per election
Triggers	The user has selected to view the election results
Preconditions	The system has completed running the election
Postconditions	Election results are displayed
Main Course	<ol style="list-style-type: none"><li>1. User selections option to view election results</li><li>2. Election results are displayed</li></ol>
Alternate Courses	AC_1: Election is in progress <ol style="list-style-type: none"><li>1. Alert user election is in progress and could take several minutes to complete</li></ol>
Exceptions	EX_1: Election hasn't been run and isn't in progress <ol style="list-style-type: none"><li>1. Alert user that they need to set up and run an election to see results</li></ol>

Name	
ID	UC_7
Description	
Actors	
Organizational Benefits	
Frequency of Use	
Triggers	
Preconditions	
Postconditions	
Main Course	1.
Alternate Courses	AC_1:
Exceptions	EX_1:

Name	The voting system provides election audit report to users
ID	UC_8
Description	The system produces a report to users. The report shows the ballots that were assigned to a candidate as the election progressed. The report is in a text file. The report does not print to the screen. All pertinent election data are included in the report at the top of the report (e.g. type of election, number of seats, number of candidates, winners, losers, etc)
Actors	Election officials, developers, testers
Organizational Benefits	A report acting as an audit trail provides election transparency to the users. Users can check how the results are devised. This report provides a way to double check election results. This report in turn increases trust in the election results.
Frequency of Use	Once every time the system runs an election
Triggers	The system starts to run an election
Preconditions	The system has all the information it needs to run an election. The system has the function of producing auditing report
Postconditions	The system generates a report. The system produces election results.
Main Course	<ol style="list-style-type: none"> <li>1. User starts the system</li> <li>2. User inputs all required information into the system</li> <li>3. User runs election</li> <li>4. System opens a report file</li> <li>5. System records ballots assigned to each candidate at each step in the report</li> <li>6. System records pertinent election results at the top of the report</li> <li>7. System closes and saves report</li> </ol>
Alternate Courses	<p>AC_1: User did not input all required information into the system</p> <ol style="list-style-type: none"> <li>3. System produces an error and prompts the user to input missing information</li> <li>4. Go to main course step 2</li> </ol>
Exceptions	EX_1: System cannot open report file

Name	
ID	UC_9
Description	
Actors	
Organizational Benefits	
Frequency of Use	
Triggers	
Preconditions	
Postconditions	
Main Course	1.
Alternate Courses	AC_1:
Exceptions	EX_1:

Name	
ID	UC_10
Description	
Actors	
Organizational Benefits	
Frequency of Use	
Triggers	
Preconditions	
Postconditions	
Main Course	1.
Alternate Courses	AC_1:
Exceptions	EX_1: