

Carlos Alvarenga (alvar357)  
Justin Koo (kooxx078)  
Michael McLaughlin (mclau361)  
Xiaochen Zhang (zhan4487)

Team 17  
10/08/18  
CSCI 5801 - 001  
Project #1 Use Cases Document

<b>Name</b>	Users specify input file
<b>ID</b>	UC_001
<b>Description</b>	Users specify the name of the file containing ballot records
<b>Actors</b>	Users
<b>Organizational Benefits</b>	Allows all ballots in a single election to be consolidated into one digital entity, eliminating the need to manage physical ballots.
<b>Frequency of Use</b>	Once per election
<b>Triggers</b>	The user runs the program via the CLI
<b>Preconditions</b>	<ul style="list-style-type: none"><li>- The specified file exists in the same directory as the program</li><li>- The program has permission to read the specified file</li></ul>
<b>Postconditions</b>	<ul style="list-style-type: none"><li>- The program reads and parses input from the specified file</li></ul>
<b>Main Course</b>	<ol style="list-style-type: none"><li>1. The user runs the program via the CLI.</li><li>2. The user specifies the name of a file that contains the desired ballot records.</li><li>3. The program confirms the specified file exists in the same directory as the program and it has permission to read the contents (see EX1).</li><li>4. The program reads and parses input from the specified file.</li></ol>
<b>Alternate Courses</b>	No alternate courses.
<b>Exceptions</b>	EX1 The program determines that either the specified file does not exist in the same directory as the program or the program does not have permission to read the contents. <ol style="list-style-type: none"><li>1. The program notifies the user of the specific error</li><li>2. Return user to Main Course step 2.</li></ol>

<b>Name</b>	Users see results
<b>ID</b>	UC_002
<b>Description</b>	Users can see the information relating to the election as well as the winner(s) on the CLI.
<b>Actors</b>	Users
<b>Organizational Benefits</b>	Increases productivity of organization by allowing them to quickly see election results rather than having to derive the results manually.
<b>Frequency of Use</b>	Mandatory. Once every successful run of the program.
<b>Triggers</b>	The program finishes the ballot tallying phase.
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- The ballot tallying phase encountered no fatal errors.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The following information is visible on the CLI: type of election, the number of candidates, the list of candidates and parties, the number of seats (OPL only), the number of ballots, and the winner(s).</li> </ul>
<b>Main Course</b>	<ol style="list-style-type: none"> <li>1. The program finishes the ballot tallying phase with no fatal errors (see EX1).</li> <li>2. Election information and its winner(s) are displayed to the CLI.</li> </ol>
<b>Alternate Courses</b>	No alternate courses.
<b>Exceptions</b>	<p>EX1 The program encounters fatal errors during the ballot tallying phase.</p> <ol style="list-style-type: none"> <li>1. The program notifies the user of the error.</li> <li>2. Return to UC_001.</li> </ol>

<b>Name</b>	Users view audit file
<b>ID</b>	UC_003
<b>Description</b>	Users can open and view the audit file generated by the program.
<b>Actors</b>	Users
<b>Organizational Benefits</b>	Allows the organization a method of validating the election results should there be a doubt on the integrity of the results.
<b>Frequency of Use</b>	Anytime the user wishes to view the audit file.
<b>Triggers</b>	The user opens the generated audit file.
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- The user is examining the directory in which the program resides.</li> <li>- The audit file exists in the directory in which the program resides.</li> <li>- The user has permission to read the contents of the audit file.</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The contents of the audit file are visible to the user.</li> </ul>
<b>Main Course</b>	<ol style="list-style-type: none"> <li>1. The user navigates to the directory in which the program resides</li> <li>2. The user attempts to open the audit file generated (see AC1)</li> <li>3. The local machine confirms that the user has permission to read the contents of the audit file (see EX2)</li> <li>4. The contents of the audit file are visible to the user</li> </ol>
<b>Alternate Courses</b>	AC1 The user determines that the audit file does not exist in the directory in which the program resides. <ol style="list-style-type: none"> <li>1. Return to UC_001</li> </ol>
<b>Exceptions</b>	EX1 The local machine determines that the user does not have permission to read the contents of the audit file. <ol style="list-style-type: none"> <li>1. The local machine notifies the user of the permission restriction.</li> <li>2. The user somehow obtains the necessary permissions.</li> <li>3. Return to Main Course step 2.</li> </ol>

<b>Name</b>	Testers run the voting program
<b>ID</b>	UC_004
<b>Description</b>	Tester types in command line arguments to run the program or to run a testing script to compile and run the voting program
<b>Actors</b>	Testers
<b>Organizational Benefits</b>	Allows the user to verify the correctness and efficiency of the program so that it's verified to be performing correctly according to the project standards
<b>Frequency of Use</b>	Whenever the user wants to test the program
<b>Triggers</b>	The user types in a command-line arguments to run the voting program or run a script that runs the program being tested
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- User is operating in a CSE lab machine</li> <li>- User is examining the directory in which the program resides</li> <li>- User has permission to write in the directory</li> <li>- Election being run is instant runoff voting or open party listing</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The election winner and the contents of the audit file are visible to the user</li> </ul>
<b>Main Course</b>	<ol style="list-style-type: none"> <li>1. The user navigates to the directory in which the program resides</li> <li>2. The user attempts to compile and run the program through command line argument "javac filename.java" or runs a testing script that'll do this job</li> <li>3. The local machine confirms that program files exist to compile and run them (see EX1)</li> <li>4. The local machine compiles the program files (see EX2)</li> <li>5. The local machine runs the program files so that user can verify the program's correctness (See AC1)</li> <li>6. The program outputs the winner on the command line and generates an audit file in the same working directory</li> </ol>
<b>Alternate Courses</b>	AC1 The local machine can't run the program or it doesn't terminate <ol style="list-style-type: none"> <li>1. Program is verified to be incorrect</li> </ol>
<b>Exceptions</b>	EX1: The local machine determines that the program files aren't in the same directory <ol style="list-style-type: none"> <li>1. The local machine notifies the user that the program files can't be found</li> <li>2. The user verifies that the program files are downloaded and placed in the same working directory</li> <li>3. Return to main course 1</li> </ol> EX2: The local machine can't compile the program files <ol style="list-style-type: none"> <li>1. User verifies that the local machine is running the most updated version of Java <ol style="list-style-type: none"> <li>a. If the correct version is being used, the program is deemed to be incorrect</li> </ol> </li> </ol>

	b. If not, update Java to its current versions and return to main course 1
--	--

<b>Name</b>	Election officials run the voting program
<b>ID</b>	UC_005
<b>Description</b>	Election official(s) run the program by providing ballot files to the voting program to determine the election winner of an instant runoff voting or open party listing voting election.
<b>Actors</b>	Election official(s)
<b>Organizational Benefits</b>	Allows the user to accurately and efficiently carry out instant runoff voting and open party listing voting elections to elect a candidate according to the public's expressed interest. By doing so, an unbiased process is implemented to determine election winners and ensuring a democratic process in society.
<b>Frequency of Use</b>	Whenever an election is being finalized
<b>Triggers</b>	The user types in a command-line arguments to run the voting program
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>- User is operating in a CSE lab machine or local machine that supports a system that has the program software integrated within it</li> <li>- User is examining the directory in which the program resides</li> <li>- User has permission to run the voting program</li> <li>- Election being run is instant runoff voting or open party listing</li> </ul>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- The election winner and the contents of the audit file are visible to the user</li> </ul>
<b>Main Course</b>	<ol style="list-style-type: none"> <li>1. The user navigates to the directory in which the program resides in</li> <li>2. The user attempts to compile and run the program through a command line argument "javac filename.java" or a GUI that hasn't been designed yet</li> <li>3. The local machine confirms that the program files exist to compile and run them (see EX1)</li> <li>4. The local machine compiles the program files (see EX2)</li> <li>5. The local machine runs the program files (See AC1)</li> <li>6. The program outputs the winner on the command line or GUI and generates an audit file in the same working directory that the program resides in</li> </ol>
<b>Alternate Courses</b>	<p>AC1 The local machine can't run the program files or it doesn't terminate</p> <ol style="list-style-type: none"> <li>1. The user makes sure that the installed software is capable of running the program</li> <li>2. Return to main course 1</li> </ol>
<b>Exceptions</b>	<p>EX1: The local machine determines that the program files aren't in the same directory or don't exist</p> <ol style="list-style-type: none"> <li>1. The local machine notifies the user that the program files can't be found</li> <li>2. The user verifies that the program files are downloaded and placed in the appropriate directory where they can be run</li> </ol>

	<ol style="list-style-type: none"><li>3. Return to main course 1</li></ol> <p>EX2: The local machine can't compile the program files</p> <ol style="list-style-type: none"><li>1. The user makes sure that the installed software is capable of running the program</li><li>2. Return to main course 1</li></ol>
--	--