Software Requirements Specification

for

<Voting System>
Version 1.0 approved
Prepared by Team#3

Shi Chen (chen4264)
Dong Liang (liang492)
Song Liu (liux4169)
Qing Hong (hong0465)

Team#3
10/8/18

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Voting System	10/7/18	Initial Specification	1.0

1. Introduction

1.1 Purpose

The overall purpose of this document is to present a detailed and precise description of the Voting System. It will describe the functionality, constraints, and all requirements of the Voting System. It will cover all the functionalities of the entire system including specific requirements for the two different election types and user interactions.

1.2 Document Conventions

The document follows the IEEE Software Requirements Specification Template.

1.3 Intended Audience and Reading Suggestions

There are 3 categories of users in this system. The first category is election officials. Election officials are the ones who will use the software to determine the winner(s) of the election. While not a user class, media personnel will be provided the results of the election and thus need an understandable summary of the results.

Programmers will use the program to unit test their functions and testers will use the program to perform integration testing.

It is recommended that all user classes read Section 1 as a briefing on the purpose of this document. Then they should proceed Section 2 for a simple overview of the product functions and user classes. Programmers and testers should read all of Section 3 carefully. Officials should focus on Section 3.1, as it describes the user interface they will be dealing with, and Section 3.3 as it describes the requirements for the software to be ran. Programmers and testers need to read Section 4, 5, and 6 as it describes in detail all functional and nonfunctional requirements. Those sections are optional for election officials.

1.4 Product Scope

The Voting System is a software that performs two different types of voting methods. The two voting systems the software will implement is the Open Party Listing (OPL) and the Instant Runoff (IR).

In Instant Runoff, candidates are ranked on a ballot. The number one preference of the ballots are counted. If a candidate has a majority of the votes, he/she is declared the winner. If there is no majority, the candidate with the least votes is eliminated and his/her ballots are redistributed to the ballots' number two candidates. This is repeated until someone has won a majority. For a more detailed overview, see Section 1.5.

In Open Party Listing, there can be one or more seats available. On a ballot, a vote for a candidate applies to the candidate and the candidate's party. Seats are allocated to the parties based on the number of votes the party received. See Section 1.5 for more information on Open Party Listing.

This system allows users to pass in a file with all ballots of an election and based on the type of voting method specified on the file, the software will determine the winner (Instant Runoff) or winner(s) and allocation of seats (Open Party Listing) and provide an audit file that the user will be able to recreate the election with. Additionally, a report of the results will be produced, and a summary will also be displayed to the user on the screen. This will provide an automated way to calculate election results as well as maintain the transparency of the election process.

1.5 References

I.E.E.E Software Requirements Specification Template:

- https://ay17.moodle.umn.edu/mod/resource/view.php?id=1347603

Detailed Explanation of Instant Runoff Voting at FairVote

- https://www.fairvote.org/plurality_majority_systems

Detailed Explanation of Open Party Listing Voting at FairVote

- https://www.fairvote.org/proportional_representation_voting_systems

2. Overall Description

2.1 Product Perspective

The Voting System will be a self-contained product that is run on a desktop. It is a new product. The long-term goal is for it to eventually be part of an integrated online voting system.

2.2 Product Functions

Functional:

Basic functions:

- Software can receive a text input in command line as file name.
- Software can read a CSV file.
- Software can analyze the comma separated data inside the CSV file.
- Software can produce an audit file with election information.

- Software can display the winner(s) and information about the election.
- Software can handle both methods of election.
- Software should randomly select the winner if there is a tie.
- Software should make the results of the election shareable and readable to media personnel.

Instant Runoff Voting

- Popularity should win after handing out all votes if there is no clear majority.
- Audit file should follow the order of ballots.
- Audit file should show all of the steps.
    - Software should show the order of the removal of candidates and how the ballots were redistributed.

Open Party Listing

- Software should treat all independents as one party.
- Audit file should follow the order of ballots.
- Audit file should show all of the steps.
    - Should show the distribution of the seats.
    - Software should calculate the winners based on number of seats and popularity of the candidates.

Non-Functional:

- An election should be able to run 100,000 ballots in under 8 minutes.
- The election file will be located in the same directory as the program.
- This program will be run multiple times during the year at normal election times and special elections.

2.3 User Classes and Characteristics

There are mainly three different user classes.

The first user class is programmers, they will not only build the product but also use the product. They are responsible for implementing the functions of the product and using the requirement documentation to specify all requirements.

The second user class is testers, they will both test the functions and use the product. They are responsible to test the product to make sure that it will follow the voting system rules and produce correct results for the voting.

The third user class is election officials, they will use the product to really run with election file, which contains all election data, and generate the voting data and result as an audit file.

2.4 Operating Environment

The system must run on a University of Minnesota CSE lab machine.

- Ubuntu Linux machine
- Windows 10 Enterprise

2.5 Design and Implementation Constraints

Runtime constraints: System should be able to run 100,000 ballots in under 8 minutes.
Coding language: Java.
Security considerations: There are no special safety or security requirements.
Maintenance: This program will be run multiple times during the year at normal election times. and special elections. Therefore, the customer's organization should be responsible for it.
Database: The election file is provided in advance and are assumed with no mistakes.

2.6 User Documentation

This Requirement Documentation will be taken as a reference for programmers and testers.

2.7 Assumptions and Dependencies

We assume different Java versions would not affect the usage of the product. We also assume different eclipse versions and configurations would not affect the compilation and running of the product. Finally, we assume the size of election file would not affect the speed of running ballots.

3. External Interface Requirements

3.1 User Interfaces

This software uses command line as user interface.

1. The software asks the user to input the address of the needed .csv file.
2. The software does the calculation and display the result in the command line.

3.2 Hardware Interfaces

There are no hardware interfaces. The audit file size is based on the size of the election.

3.3 Software Interfaces

This software requires Java to be installed on the system, more specifically Java version 7 or 8 for its latest release. This software may need Excel to be installed on the system.

3.4 Communications Interfaces

This software may need the Internet connection to validate the username and password and upload the final voting result.

4. System Features and Functional Requirements

4.1 System Prompts User for Filename

4.1.1 Description and Priority

The program must know the name of the csv of the file, so the user is prompted for the name of the csv file. See UC_001, UC_002, UC_011, and UC_012 in Appendix D for detailed use cases split into two different voting methods and for officials and testers. This is a high priority function as it allows the rest of the program to function properly.

4.1.2 Stimulus/Response Sequences

1. User is asked to provide the filename.
   Stimulus: The program has been compiled.
   Response: Program displays a text prompt on terminal asking for the name of the csv file.

2. User provides filename.
   Stimulus: User types in filename and hits enter.
   Response: Program reads in the csv file specified by the name.

4.1.3 Functional Requirements

REQ-1: User Needs to Pass in an Instant Runoff election filename (see UC_001 and UC_011 in Appendix D).
Details and Constraints:
System needs to prompt the user for a file name to be read into a system. The file must be in the same directory as the program. The file must be a CSV. Instant Runoff files must have “IR” written on the first line. Each election is confined to the one file.

REQ-2: User Needs to Pass in Open Party Listing election filename (see UC_002 and UC_012 in Appendix D).
Details and Constraints:
System needs to prompt the user for a file name to be read into a system. The file must be in the same directory as the program. The file must be a CSV. Open Party Listing files must have “OPL” written on the first line. Each election is confined to the one file.

4.2 User Runs Instant Runoff Election

4.2.1 Description and Priority

User inputs a file with ballots from an Instant Runoff election and the system produces the election results, a summary report, and an audit report. See UC_003 and UC_013 in Appendix D for official and tester use cases. This is a high priority requirement and is one of the two main functions required for the officials.

4.2.2 Stimulus/Response Sequences

1. User provides a CSV file of Instant Runoff ballots and receives the results of the election.
   Stimulus: User provides a CSV file with “IR” on its first line.
   Response: The system performs the Instant Runoff voting method and the

corresponding election summary is displayed to the user on the screen, a summary report is written to the root directory, and an audit report is written to the root directory.

4.2.3 Functional Requirements

REQ-1: System Perform an Instant Runoff Election (see UC_003 and UC_013 in Appendix D).
Details and Constraints: If the user passes in a csv file with “IR” written on the first line, the system performs an Instant Runoff election.
Error Conditions or Invalid Inputs: There will be no ballot or file input errors and all ballots will have at least one candidate ranked.

REQ-1.1: System Break Tie in Instant Runoff.
Details and Constants: If there is no majority at the end of the election, the most popular member is the winner. If there is a tie in the popular vote, then pick the winner from the tied candidates randomly.

REQ-2: System Generates an Audit Report of Instant Runoff Election for the User (see UC_009 in Appendix D).
Details and Constraints: An audit file of the Instant Runoff election is written and placed in the same directory as the input file. There are no size or file type constraints.

REQ-2.1: The Audit Report Must Provide Basic Information for the User.
Details and Constraints: The basic information includes the type of method (Instant Runoff), the number of candidates and their names, the number of votes each candidate received, the number of ballots cast, and the winner.

REQ-2.2: The Audit Report Must Show the Entire Process of an Instant Runoff Election.
Details and Constraints: The file should show how the election progressed so that the audit could replicate the election itself. It needs to show the order of the ballots being assigned to each candidate. This means the ballots must be time stamped to show the time of distribution. It also needs to show the order of the candidates being removed and the redistribution of those candidates’ votes.

REQ-3: System Generates Summary Report of an Instant Runoff Election for the User (see UC_005).
Details and Constraints: A brief summary of the Instant Runoff election results is written and placed in the same directory as the input file. The summary should include the election type, winner of the election, the number of ballots cast, and the number of votes each candidate received in each round.

REQ-4: System Displays Results of Instant Runoff Election to the User (see UC_007).
A brief summary of the Instant Runoff election results is displayed to the user on the terminal screen. The summary should include the election type, winner of the election, the number of ballots cast, and the number of votes each candidate received in each round.

4.3 User Runs Open Party Listing Election

4.3.1 Description and Priority

User inputs a file with ballots from an Open Party Listing election and the system produces the election results, a summary report, and an audit report. See UC_004 and UC_014 in Appendix D for official and tester use cases. This is a high priority requirement is one of the two main functionalities required for the officials.

4.1.2 Stimulus/Response Sequences

1. User provides a csv file of Open Party Listings ballots and receives the results of the election.
   Stimulus: User provides a CSV file with “OPL” on its first line.
   Response: The system performs the Open Party Listing voting method and the corresponding election summary is displayed to the official on the screen, a summary report is written to the root directory, and an audit report is written to the root directory.

4.1.3 Functional Requirements

REQ-1: System Perform an Instant Runoff Election for the User (see UC_004 and UC_014 in Appendix D).
Details and Constraints: If the user passes in a CSV file with “OPL” written on the first line, the system performs an Open Party Listing election.
Error Conditions or Invalid Inputs: There will be no ballot or file input errors and there will be at least as many candidates as the number of seats.

REQ-1.1: System Needs to Break Ties in Open Party Listing Election.
Details and Constants: Any ties in the distribution of seats to the party or to the candidates must be broken randomly.

REQ-2: System Must Generate an Audit Report of an Open Party Election for the User (see UC_010 in Appendix D).
Details and Constraints: An audit file of the Open Party Listing election is written and placed in the same directory as the input file. There are no size or file type constraints.

REQ-2.1: The Audit Report Must Provide Basic Information to the User.
Details and Constraints: The basic information includes the type of method (Open Party Listing), the number of candidates and their names and the number of votes each candidate received, the number of seats, the number of ballots cast, and the winner(s).

REQ-2.2: The Audit Report Must Show the Entire Process of an Open Party Listing to the User.
Details and Constraints: The file should show how the election progressed so that the audit could replicate the election itself. It needs to show the order of the ballots being assigned to each candidate. This means the ballots must be time stamped to show the time of distribution. It also needs to show how the distribution of the seats to the parties is calculated. Finally, the must show how the parties’ seats are distributed to the winning candidates.

REQ-3: System Generate Summary Report of an Open Party Listing Election for the User (see UC_006 in Appendix D).
Details and Constraints: A brief summary of the Instant Runoff election results is written and placed in the same directory as the input file. The summary should include the election type, winner of the election, the number of ballots cast, and the number of votes each candidate received in each round.

REQ-4: System Displays Results of Instant Runoff Election to the User (see UC_008 in Appendix D).
A brief summary of the Instant Runoff election results is displayed to the user on the terminal screen. The summary should include the election type, winner of the election, the number of ballots cast, and the number of votes each candidate received in each round.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. Performance requirement (Runtime constraints):

- An election should be able to run 100,000 ballots in under 8 minutes.

2. Frequency requirement:

- This program will be run multiple times during the year at normal election times and special elections.

3. Special condition requirement (Tie problem):

- If there is ever a tie, flip a coin. The program will randomly select the winner in a fair coin toss.

5.2 Safety Requirements

There are no special safety or security requirements. Security such as ensuring one vote for one person is handled at the voting centers.

5.3 Security Requirements

There are no special safety or security requirements. Security such as ensuring one vote for one person is handled at the voting centers.

5.4 Software Quality Attributes

For developers, programmers, testers, and election officials:

- The results of the election could be shared with media personnel.

For correctness:

- The only concern is when a tie happens, the system will pick a random winner instead of doing extra handlings.

For reliability and testability:

- The system will generate 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.). The system should list the winner(s), and you should show how the election progressed so that the audit could replicate the election itself. Unless there is a tie, the election should produce the same winner(s) for the same file if ran multiple times.

5.5 Business Rules

Users can assume that they will use the most up-to-date CSE Labs machines (This system is guaranteed to be runnable on a CSE Labs machine).

The file cannot be changed outside of the program.

6. Other Requirements

There will be no write in candidates for this system. It may happen in the future but not now.

Appendix A: Glossary

CSV: **C**omma **S**eparated **V**alues file format.
OPL: **O**pen **P**arty **L**isting voting method.
IR: **I**nstant **R**unoff voting method.

Appendix B: Analysis Models

Will be added later.

Appendix C: To Be Determined List

Analysis models must be added later.

Appendix D: Use Cases

UC_001

Name	Official passes name of an Instant Runoff file to program.
ID	UC_001
Description	We prompt the user for the file name. A terminal input of the file name is typed in by the user.
Actors	Election Officials
Organizational Benefits	The election officials are able to determine the winner of an election without manual calculations.
Frequency of Use	Multiple times during an election season or special elections.
Triggers	Program is built and active.
Preconditions	1. Election file is in the root directory as the program. 2. Election file is marked “IR” on 1st line. 3. File is a CSV format.
Post Conditions	1. The file is read into the program and ready for the Instant Runoff method.
Main Course	1. Program asks user for file name using string output on the terminal. 2. User types in file name and hit enter.
Alternate Courses	None

Exceptions	This iteration will have no file or input errors.

UC_002

Name	Official passes name of an Open Party Listing file to program.
ID	UC_002
Description	We prompt the user for the file name. A terminal input of the file name is typed in by the user.
Actors	Election Officials
Organizational Benefits	The election officials are able to determine the winner of an election without manual calculations.
Frequency of Use	Multiple times during an election season or special elections.
Triggers	Program is built and active.
Preconditions	1. Election file is in the root directory as the program. 2. Election file is marked “OPL” on 1st line. 3. File is a CSV format.
Post Conditions	The file is read into the program and ready for the Open Party Listing voting method.
Main Course	1. Program asks user for file name using string output on the terminal. 2. User types in file name and hits enter.
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_003

Name	Election Officials run the Instant Runoff voting method.
ID	UC_003
Description	Election Official passes in a file of an Instant Runoff election and the election results are generated.
Actors	Election Officials
Organizational Benefits	Election results can be calculated without manual effort.
Frequency of Use	Multiple times during an election season or special elections.

Triggers	User passes in a CSV file name to the program (see UC_001).
Preconditions	File is in the same directory as the program. File has “IR” written on the first line.
Post Conditions	1. An audit file of the Instant Runoff is available to the user in the same directory as the input file. 2. A summary file of the Instant Runoff is available to the user in the same directory as the input file.
Main Course	1. System performs Instant Runoff voting. 2. An audit file of the Instant Runoff election is written to the directory of the program and visible to official in the directory. 3. Official views summary results on terminal screen. 4. A summary report is written to directory of the program and visible to the official in the directory.
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_004

Name	Election Officials run the Open Party Listing voting method.
ID	UC_004
Description	Election Official passes in a file of an Open Party Listing election and the election results are generated.
Actors	Election Officials
Organizational Benefits	Election results can be calculated without manual effort.
Frequency of Use	Multiple times during an election season or special elections.
Triggers	User passes in a csv file name to the program (see UC_002).
Preconditions	1. File is in the same directory as the program. 2. File has “OPL” written on first line
Post Conditions	1. An audit file of the Open Party Listing election is available to the user in the same directory as the input file. 2. A summary file of the Open Party Listing is available to the user in the same directory as the input file (see UC_005).
Main Course	1. System performs Open Party Listing voting. 2. An audit file of the Open Party Listing election is written to the

	directory of the program and visible to official in the directory. 3. Official views summary results on terminal screen. 4. Summary report is written to directory of the program and visible to the official in the directory.
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_005

Name	Generate Instant Runoff election summary reports.
ID	UC_005
Description	When an Election Official runs an Instant Runoff election, a summary report is generated. It contains summary results like the type of the election, the winners, and number seats, etc.
Actors	1. Election Official Related: Media personnel will receive the election results.
Organizational Benefits	The results of an Instant Runoff election are summarized succinctly and clearly.
Frequency of Use	Multiple times during an election season or special elections.
Triggers	User passes in a csv file name to the program (see UC_001).
Preconditions	1. Election file is in the same directory as the program. 2. Election file has “IR” on line 1.
Post Conditions	A summary report of the Instant Runoff election is in the same directory as the election file.
Main Course	1. System performs Instant Runoff voting. 2. A summary report is written to the same directory as the election file and visible to the election official and accessible to the official. File should show the: • Type of election • Number of ballots cast • Number of votes received for each candidate. • The winner of the election.
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_006

Name	Generate Open Party Listing election summary reports.
ID	UC_006
Description	When an Election Official runs an Open Party Listing election, a summary report is generated. It contains summary results like the type of the election, the winners, and number seats, etc.
Actors	1. Election Official Related: Media personnel will receive the election results.
Organizational Benefits	The results of an Open Party Listing election are summarized succinctly and clearly.
Frequency of Use	Multiple times during an election season or special elections.
Triggers	User passes in a csv file name to the program (see UC_001).
Preconditions	1. Election file is in the same directory as the program. 2. Election file has "OPL" on line 1.
Post Conditions	A summary report of the Open Party Listing election is in the same directory as the election file.
Main Course	1. System performs Open Party Listing voting. 2. A summary report is written to the same directory as the election file and accessible to the user. File should show the: • Type of election • Number of seats in the election • Number of ballots cast • Number of votes received for each candidate • Winners of the election • Seats won by each party
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_007

Name	Display a summary of an Instant Runoff election to the Election Official on the screen.
ID	UC_007
Description	When an Election Official runs an Instant Runoff election, a summary of the

	election is displayed to the user with information like the winners, the type of the election, etc.
Actors	Election Official
Organizational Benefits	The results of an Instant Runoff election are summarized and displayed succinctly and clearly.
Frequency of Use	Multiple times during an election season or special elections.
Triggers	User passes in a csv file name to the program (see UC_001).
Preconditions	1. Election file is in the same directory as the program. 2. Election file has "IR" on line 1.
Post Conditions	A summary report of the Instant Runoff election is displayed on the terminal.
Main Course	1. System performs Instant Runoff voting. 2. A summary report is displayed to the election official on the screen. It should show: • Type of election • Number of ballots cast • Number of votes received for each candidate. • Winner of the election
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_008

Name	Display a summary of an Open Party Listing election to the Election Official on the screen.
ID	UC_008
Description	When an Election Official runs an Instant Runoff election, a summary of the election is displayed to the user with information like the winners, the type of the election, the number of seats won by each party, etc.
Actors	Election Official
Organizational Benefits	The results of an Open Party Listing election are summarized and displayed succinctly and clearly.
Frequency of Use	Multiple times during an election season or special elections.
Triggers	User passes in a csv file name to the program (see UC_001).

Preconditions	1. Election file is in the same directory as the program. 2. Election file has “OPL” on line 1.
Post Conditions	A summary report of the Open Party Listing voting is displayed on the screen.
Main Course	1. System performs Open Party Listing voting. 2. A summary report is displayed to the election official on the screen. It should show: • Type of election • Number of ballots cast • Number of votes received for each candidate. • Winners of the election • Number of seats in the election
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_009

Name	Election official receives an audit report for an Instant Runoff election.
ID	UC_009
Description	When an Election Official runs an Instant Runoff election, an audit report is generated with details such as the winner and how the election progressed.
Actors	Election Official
Organizational Benefits	Helps ensure transparency of the election process.
Frequency of Use	Multiple times during an election season or special elections.
Triggers	User passes in a csv file name to the program (see UC_001).
Preconditions	1. Election file is in the same directory as the program. 2. Election file has “IR” on line 1.
Post Conditions	An audit report of the Instant Runoff election is in the same directory as the election file.
Main Course	1. System performs Instant Runoff voting. 2. An audit report is written to the directory of the election file and accessible to the user. It should show: • Instant Runoff Election • Number of ballots cast

	• Number of votes received for each candidate. • Winner of the election • Calculations to determine the winner. • Progression of the election
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_010

Name	Election official receives an audit report for an Open Party Listing election.
ID	UC_010
Description	When an Election Official runs an Open Party Listing election, an audit report is generated with details such as the winner and how the election progressed.
Actors	Election Official
Organizational Benefits	Helps ensure transparency of the election process.
Frequency of Use	Multiple times during an election season or special elections.
Triggers	User passes in a csv file name to the program (see UC_001).
Preconditions	1. Election file is in the same directory as the program. 2. Election file has "OPL" on line 1.
Post Conditions	An audit report of the Open Party Listing election is in the same directory as the election file.
Main Course	1. System performs Instant Open Party Listing voting. 2. An audit report is written to the directory of the election file and accessible to the user. It should show: • Open Party Listing • Number of candidates and their names • Number of ballots cast • Number of votes received for each candidate. • Winners of the election • Calculations to determine the winners. • Progression of the election.
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_011

Name	Programmer tests inputting Instant Runoff file name.
ID	UC_011
Description	The programmer should be able to test the inputting of the file name for an Instant Runoff election.
Actors	Programmer
Organizational Benefits	The correctness of the system is ensured.
Frequency of Use	Frequently in the Implementation and Verification Stages
Triggers	Program is built and active.
Preconditions	4. Election file is in the root directory as the program. 5. Election file is marked “IR” on 1st line. 6. File is a csv format.
Post Conditions	1. File is read into the program.
Main Course	1. Program asks programmer for file name using string output on the terminal. 2. Programmer types in file name and hit enter. 3. Programmer verifies file has been read properly.
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_012

Name	Programmer tests inputting Open Party Listing file name.
ID	UC_012
Description	The programmer should be able to test the inputting of the file name for an Open Party Listing election.
Actors	Programmer
Organizational Benefits	The correctness of the system is ensured.
Frequency of Use	Frequently in the Implementation and Verification Stages
Triggers	Program is built and active.

Preconditions	1. Election file is in the root directory as the program. 2. Election file is marked “OPL” on 1st line. 3. File is a csv format.
Post Conditions	1. File is read into the program.
Main Course	1. Program asks programmer for file name using string output on the terminal. 2. Programmer types in file name and hit enter. 3. Programmer verifies file has been read properly.
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_013

Name	Programmer or tester tests Instant Runoff election.
ID	UC_013
Description	The tester should be able to test the correctness of the Instant Runoff election
Actors	Testers and programmers
Organizational Benefits	The correctness of the system is ensured.
Frequency of Use	Frequently in the Implementation and Verification Stages
Triggers	Tester or programmers passes in an Instant Runoff file name for testing.
Preconditions	1. Election file is in the root directory as the program. 2. Election file is marked “IR” on 1st line. 3. File is a csv format.
Post Conditions	1. Audit and summary file are in file directory. 2. Summary results are displayed to the screen.
Main Course	1. System performs Instant Runoff voting. 2. Tester/programmer verifies correctness of visual summary. 3. Tester/programmer verifies correctness of audit report. 4. Tester/programmer verifies correctness of summary report.
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.

UC_014

Name	Programmer or tester tests Open Party Listing election.
ID	UC_014
Description	The tester or programmer should be able to test the correctness of the Open Party Listing election.
Actors	Testers and programmers
Organizational Benefits	The correctness of the system is ensured.
Frequency of Use	Frequently in the Implementation and Verification Stages
Triggers	Tester or programmers passes in an Open Party Listing file name for testing.
Preconditions	1. Election file is in the root directory as the program. 2. Election file is marked "OPL" on 1st line. 3. File is a csv format.
Post Conditions	1. Audit and summary file are in file directory. 2. Summary results are displayed to the screen.
Main Course	1. System performs Open Party Listing voting. 2. Tester/programmer verifies correctness of visual summary. 3. Tester/programmer verifies correctness of audit report. 4. Tester/programmer verifies correctness of summary report.
Alternate Courses	None
Exceptions	This iteration will have no file or input errors.