# Software Requirements Specification

# for

# Party List Voting System

Version 1.0 approved

Prepared by Wong. A(wang8039), Olson. H (olso7397), Li,G(li000490), Koversky. A(kover002)

CSCI 5081 Group 10

October 10, 2019

Table of Contents

Table of Contents [ii](#__RefHeading___Toc441230970)

Revision History [ii](#__RefHeading___Toc441230971)

1. Introduction [1](#__RefHeading___Toc441230972)

1.1 Purpose [1](#__RefHeading___Toc441230973)

1.2 Document Conventions [1](#__RefHeading___Toc441230974)

1.3 Intended Audience and Reading Suggestions [1](#__RefHeading___Toc441230975)

1.4 Product Scope [1](#__RefHeading___Toc441230976)

1.5 References [1](#__RefHeading___Toc441230977)

2. Overall Description [2](#__RefHeading___Toc441230978)

2.1 Product Perspective [2](#__RefHeading___Toc441230979)

2.2 Product Functions [2](#__RefHeading___Toc441230980)

2.3 User Classes and Characteristics [2](#__RefHeading___Toc441230981)

2.4 Operating Environment [2](#__RefHeading___Toc441230982)

2.5 Design and Implementation Constraints [2](#__RefHeading___Toc441230983)

2.6 User Documentation [2](#__RefHeading___Toc441230984)

2.7 Assumptions and Dependencies [3](#__RefHeading___Toc441230985)

3. External Interface Requirements [3](#__RefHeading___Toc441230986)

3.1 User Interfaces [3](#__RefHeading___Toc441230987)

3.2 Hardware Interfaces [3](#__RefHeading___Toc441230988)

3.3 Software Interfaces [3](#__RefHeading___Toc441230989)

3.4 Communications Interfaces [3](#__RefHeading___Toc441230990)

4. System Features [4](#__RefHeading___Toc441230991)

4.1 Process Closed Party List [4](#__RefHeading___Toc441230992)

4.2 Process Open Party List [4](#__RefHeading___Toc441230993)

4.3 Create Audit Log [4](#__RefHeading___Toc441230992)

4.4 Update Audit Log [4](#__RefHeading___Toc441230993)

4.5 Create Media Summary [4](#__RefHeading___Toc441230992)

4.6 Read In Input File [4](#__RefHeading___Toc441230993)

5. Other Nonfunctional Requirements [4](#__RefHeading___Toc441230994)

5.1 Performance Requirements [4](#__RefHeading___Toc441230995)

5.2 Safety Requirements [5](#__RefHeading___Toc441230996)

5.3 Security Requirements [5](#__RefHeading___Toc441230997)

5.4 Software Quality Attributes [5](#__RefHeading___Toc441230998)

5.5 Business Rules [5](#__RefHeading___Toc441230999)

6. Other Requirements [5](#__RefHeading___Toc441231000)

Appendix A: Glossary [5](#__RefHeading___Toc441231001)

Appendix B: Analysis Models [5](#__RefHeading___Toc441231002)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| **PLVS** | 10/10 | Initial Write-up | 0.0 |
|  |  |  |  |

# Introduction

## Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

The purpose of this document is to present a detailed description of the Voting System. It will explain the purpose of the system, system interfaces, what the system can do, and the constraints it operates under. This document is for potential stakeholders, users, and developers of the system.

## Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

This document follows the IEEE template for System Requirement Specification Documents.

## Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

This document is intended for:

* Users who are looking to implement or adopt specific aspects of the system.
* Developers who are planning to build from the system or fix present errors or bugs.

## Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

The Voting System is a tool that takes a file containing the ballots of Open Party List and Closed Party List votes and determines a winner. An audit file is produced along with the election results in case an audit of the election is required. Additionally, a file containing general election result information that may be given to media outlets for broadcasting is produced. Automating this process increases ballot worker efficiency and reduces the probability of human error interfering with vote counts.

Automating the voting process increases ballot worker efficiency and reduces the probability of human error for voting results.

## References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

Wiegers, K. (1999). IEEE Software Requirements Specification Template. Retrieved from:

https://web.cs.dal.ca/~hawkey/3130/srs\_template-ieee.doc

# Overall Description

## Product Perspective

The purpose of the PLVS software is to reduce the time needed and human error involved in the tallying of ballots for United States elections. Currently a "plurality" algorithm is used in many US elections where each voter votes for a single candidate, and the candidate with the most total votes wins. However, this is not the only voting algorithm, and it does have a serious flaw in that it doesn't directly handle ties. "Party-list" voting (PLV) allows political parties to select a candidate for each electoral district, and allows each voter to vote for a party. Political seats are then assigned based the proportion of the vote each party receives and the order each parties candidates are listed in.

The PLVS software supports two sub-categories of party-list voting, another voting algorithm used by many other democracies across the world that handles ties by allowing voters to vote for parties rather than just candidates. "Closed-party-list" voting (CPLV) allows each party to decide the order of their candidates, and each voter to vote for a party. "Open-party-list" voting (OPLV) allows the voter to pick a candidate (and their party), and determines the order of the candidates from the results of the vote. This software will allow a user to input a file containing ballots and will process them to declare a winner. It will also create an audit log for the election, and give the option to create a media summary of the results. Once fully tested and release, long term maintenance will be handled by other developers.

## Product Functions

Input:

* Input File: Allows the user to specify a file name, and loads in that specific file as the input.

Vote Processing:

* Process Input: Checks the input file to determine what type of voting algorithm should be used to evaluate the input file, and than switches over to one of the following functions based on that determination.
* Process CPLV: Allows the user to process an input file using the closed-party-list variant of party-list voting.
* Process OPLV: Allows the user to process an input file using the open-party-list variant of party-list voting.
* Break Tie: In the event of a tie, this function determines the winner through a fair random number generation.

Output:

* Display Results: Allows the user to view the winner(s) of an election after the fact, along with the total number of votes cast, the percentage of the vote that went to each candidate and party, and the number of seats that went to each party. The way this information is displayed will differ slightly based on which algorithm is used.
* Print Audit Log: This function automatically prints an audit log detailing how each individual ballot impacted an election, and how any tie-breakers were determined. This log is output into a text file for later review.
* Share results: Allows the user to generate a report summarizing the results of an election to be shared with media correspondents.

## User Classes and Characteristics

* Programmers, the developers of this software.
* Testers, who verify that the software is producing the correct results. Testers will use the audit log features extensively. There may be some overlap between the tester, programmer, and government official user classes.
* Government Officials, who will conduct the elections once the software is in use.
* Media Professionals, who will report the results of the election to the general public. Media Professionals will not use the software themselves, instead receiving a report generated by the software and delivered by the its primary users.

## Operating Environment

The PLVS software is designed to operate on the CSELabs machines. Specifically, these machines use Linux release number 4.15 as an operating system with a x86\_64 machine architecture and have 500 GB of storage. The PLVS software requires the permissions to read in the input file, and to create and save audit logs.

## Design and Implementation Constraints

* The PLVS software will be developed in Java. All source files needed to run and compile it will be included, and it will be able to be run from the command line.
* The system will be able to process 100,000 votes in five minutes or less.
* The audit log will be output to a plain-text file, and needs to be able to completely recreate the election so that results can be verified.
* The files input to the system will be passed in as CSV files. These files will start with a header portion containing the type of voting, number of seats available, number of ballots, number of candidates, and each of the candidates listed. Each of the ballots will then be expressed, and a "end of file" tag will conclude them. These input files are assumed to be error-free.
* Once delivered to the government, their organization will be responsible for maintaining it.
* There are no security constraints surrounding the voting system.

## User Documentation

A complete user guide containing how-to tutorials for each of the Voting System's features written for non-technician government officials and software design documentation for future developers will be delivered along with the software.

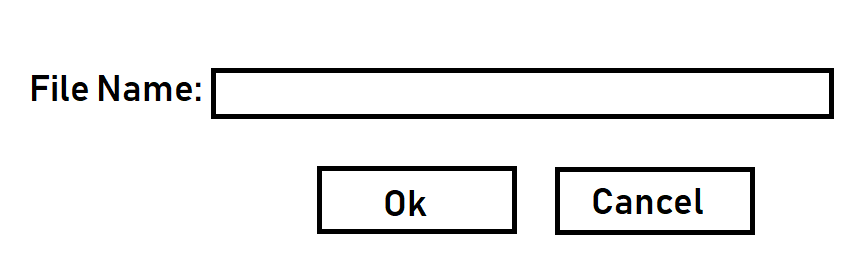
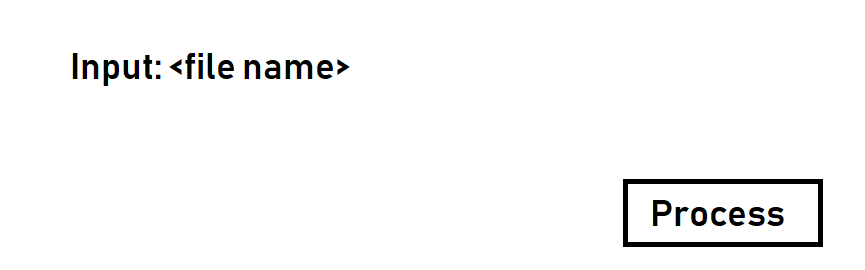
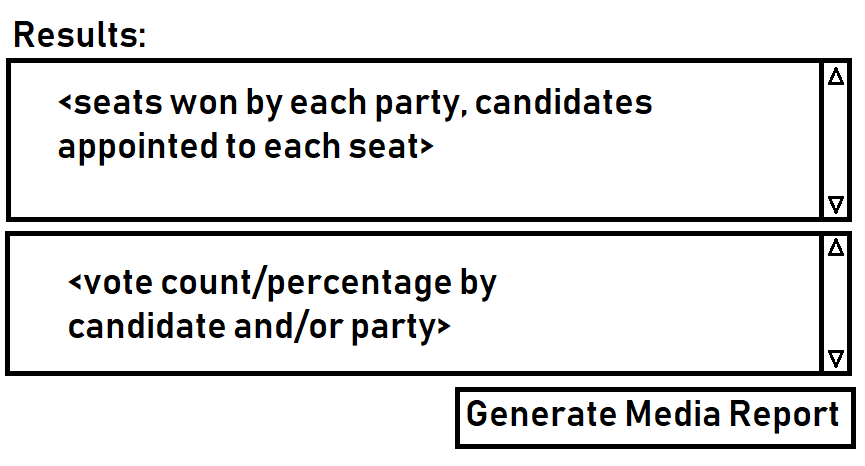
## Assumptions and Dependencies

This Voting System software will be programmed in Java, and therefore requires the Java run-time environment (JRE) to be installed on the user's computer. It is assumed that all of the Java libraries included in the CSELabs machines are present on the users computer, and that the user's computer has the permissions needed to save files and create new folders. The input file is assumed to contain no errors, and to contain at least one ballot. It is assumed that there will be no more than one input file per election.

# External Interface Requirements

## User Interfaces

Users that have logged into their accounts, after that it will pop up a GUI window prompt the user to input a filename which they want to open. And it starts read the data in the file and process it to output a audit file. And then display to the screen the winner and election information.

1. Prompt user to enter a filename.
2. Process the data in the file.
3. Generate media report.

## 3.2 Hardware Interface

The device type need to be supported:

PC Model: Dell Optiplex 9020

Linux Version: 4.15.0-65-generic

Hardware Arc: x86 / 64

Processor type: Intel i5 4570 3.2 GHz

RAM: 4GB

Storage: 500GB

## Software Interfaces

There are two software interfaces which are Java Runtime Environment and Linux’s file input and output.

## Communications Interfaces

An internet connection is required for the Party List Voting System server so it is accessible from the internet. Users trying to use Party List Voting System will require an internet connection.

# System Features

## Process Closed Party List

4.1.1 Description and Priority

The processing of closed party list inputs allows the user to automatically determine the results of an election using the CPLV algorithm. Using this algorithm, each ballot contains a vote for a single party, and the winners of the election are first candidates in the party-determined list of running candidates proportional to the percentage of the total vote captured by that party. This is a core function of the PLVS software and is of the highest priority.

4.1.2 Stimulus/Response Sequences

This feature begins when the current user clicks the "Process Votes" button and the input file indicates that the election should be processed using the CPLV algorithm.

If there is no input file, the system should alert the user of this and halt on this feature. If there is one, the system reads in the number of available seats, the number of parties, the number of ballots cast, and the ordered lists of candidates' names and affiliations from the input file. It will then tally the vote of each ballot for the correct party, and once all ballots have been processed determine the proportion of seats each party has collected. The first X candidates of each party will then be declared "winners", where X is the number of seats each party earned.

4.1.3 Functional Requirements

REQ-1: The system needs to read in the input file.

REQ-2: The system needs to determine that the election is uses CPLV from the input file.

REQ-3: The system needs to read in the number of seats, parties, ballots, candidates' names and affiliations, and the ballots themselves from the input file.

REQ-4: The system needs to tally the vote for each party as each ballot is processed

REQ-5: The system needs to update the audit log as each ballot is processed (see TBD)

REQ-6: The system needs to determine and display the results of the election once all ballots are processed (See TBD)

REQ-7: Ties should be broken using random number generation (See TBD)

REQ-8: Processing the votes can not impact the input file itself.

4.1.4 Non-Functional Requirements

REQ-1: The system read and write accesses on the user's computer.

REQ-2: There can not be any errors in the input file (ie. bad formatting, more than one vote per ballot).

REQ-3: Input files must contain at least one ballot

REQ-4: 100,000 ballots need to be processed in less than 5 minutes.

## Process Open Party List

4.2.1 Description and Priority

The processing of open party list inputs allows the user to automatically determine the results of an election using the OPLV algorithm. Using this algorithm, each ballot contains a vote for a single candidate, and the winners of the election are the X candidates in each party who receive the most votes where X is proportional to the percentage of the total vote captured by that party. This is a core function of the PLVS software and is of the highest priority.

4.2.2 Stimulus/Response Sequences

This feature begins when the current user clicks the "Process Votes" button and the input file indicates that the election should be processed using the OPLV algorithm.

If there is no input file, the system should alert the user of this and halt on this feature. If there is one, the system reads in the number of available seats, the number of parties, the number of ballots cast, and the ordered lists of candidates' names and affiliations from the input file. It will then tally the vote of each ballot for the correct candidate, and once all ballots have been processed determine the proportion of seats each party has collected. The highest voted X candidates of each party will then be declared "winners", where X is the number of seats each party earned.

4.2.3 Functional Requirements

REQ-1: The system needs to read in the input file.

REQ-2: The system needs to determine that the election is uses OPLV from the input file.

REQ-3: The system needs to read in the number of seats, parties, ballots, candidates' names and affiliations, and the ballots themselves from the input file.

REQ-4: The system needs to tally the vote for each candidate as each ballot is processed

REQ-5: The system needs to update the audit log as each ballot is processed (see TBD)

REQ-6: The system needs to determine and display the results of the election once all ballots are processed (See TBD)

REQ-7: Ties should be broken using random number generation (See TBD)

REQ-8: Processing the votes can not impact the input file itself.

4.2.4 Non-Functional Requirements

REQ-1: The system read and write accesses on the user's computer.

REQ-2: There can not be any errors in the input file (ie. bad formatting, more than one vote per ballot).

REQ-3: Input files must contain at least one ballot

REQ-4: 100,000 ballots need to be processed in less than 5 minutes.

## Create Audit Log

4.3.1 Description and Priority

<xyz>

4.3.2 Stimulus/Response Sequences

<xyz>

4.3.3 Functional Requirements

REQ-1: <xyz>

REQ-2: <xyz>

4.3.4 Non-Functional Requirements

REQ-1: <xyz>

REQ-2: <xyz>

## Update Audit Log

4.4.1 Description and Priority

<xyz>

4.4.2 Stimulus/Response Sequences

<xyz>

4.4.3 Functional Requirements

REQ-1: <xyz>

REQ-2: <xyz>

4.4.4 Non-Functional Requirements

REQ-1: <xyz>

REQ-2: <xyz>

## Create Media Summary

4.5.1 Description and Priority

When a government official or system operator selects an instruction to calculate the election result in the GUI system, the system will calculate the election result based on the result of the voting and send it to the media according to the type of data related to the election desired by the user. This is the core function of the PLVS software and has the highest priority.

4.5.2. Stimulus/Response Sequences

This function begins when the current user clicks the "Generate Poll Results" button and the input file indicates that the election should be processed using the PLVS algorithm. If there is no source file (the result of the vote), the system should warn the user and prompt the user to upload the correct file first. If the operation is successful, the system will read the type of open or close from the input file: if it is open type, then the final elected member should be calculated according to the number of votes and the ordered list of candidate names and affiliations; The type of close will determine the final elected member according to the chosen party, which is the first two members of the party. The candidate for each party to receive the highest X vote will be declared the “winner”, where X is the number of seats each party receives. The resulting results will be edited in the format required by the customer and delivered to the media.

4.5.3 Functional Requirements

REQ-1: The system receives order to create media summary

REQ-2: The system needs to determine that the election in the input file uses OPL or CPL.

REQ-3: The system needs to read seats, parties, votes, candidate names and affiliations, and the number of votes themselves from the input file.

REQ-4: The system will determine the final list of candidates based on different types of election rules.

REQ-5: The system creates final poll result.

4.5.4 Non-Functional Requirements

REQ-1: The system read and write accesses on the user's computer.

REQ-2: There can not be any errors in the input file (ie. bad formatting, more than one vote perballot).

REQ-3: Input files must contain at least one ballot

REQ-4: 100,000 ballots need to be processed in less than 5 minutes.

REQ-5: Ensuring one vote for one person is handled at the voting centers.

## Read In Input File

4.6.1 Description and Priority

When the system pops up a GUI window for entering a filename, the user entered the filename then the system will open the file, which is the vote OPL and CPL file will be exported from Excel into the CSV format, it will be comma separated values for the ballots where each row is separated by a newline. System will read the data inside the files.

4.6.2 Stimulus/Response Sequences

This feature begins when the current user clicks the “Read file” button and a GUI window pop up for entering a filename. If there is no input file or cannot find the input filename in tis directory, the system should alert the user of this and halt on this feature.

4.6.3 Functional Requirements

REQ-1: A GUI window will pop up prompting for entering a filename.

REQ-2: The user needs to input the correct file name which they want to open.

REQ-3: The system needs to open the input file.

REQ-4: The input file is readable.

REQ-5: The input file data is not altered.

4.6.4 Non-Functional Requirements

REQ-1: There cannot be any structure errors in the input file.(ie: bad formatting, more than one vote per ballot)

REQ-2: The input filename can be searched in the system successfully.

REQ-3: The input file is in the same directory as the program.

# Other Nonfunctional Requirements

## Performance Requirements

*5.1.1 Response time:*

*1. When the network is unblocked, the time required to dial-up to the GPRS network must not exceed 5 seconds.*

*2. The PLVS system should guarantee the general time response time does not exceed 1.5 seconds, at the peak time does not exceed 4 seconds in 95% of cases,*

*3. The time required for the positioning system to display from the click to the first interface must not exceed 300 milliseconds.*

*4. Search during the off-peak hours according to the specific conditions of the number and name, and the search results can be obtained within 3 seconds.*

*5.The login response time is within 2 seconds, the response time of the refresh column is within 2 seconds, the response time of the entry page list is refreshed within 2 seconds, the response time of the information entered is opened within 1 second, and the response time of the personnel list is refreshed within 2 seconds.*

*5.1.2 Flexibility and expandability*

*The flexibility of the PLVS to handle multiple tasks and perform individual tasks in different ways. That is, the flexibility to change the way this task is handled. The flexible and personalized system can truly change according to the needs of users, and always meet the needs of users in different development stages and different enterprise information management. PLVS should meet the requirements of multi-party open and closed ballot statistics, if government officials need to add new requirements, such as limiting the number of tickets, the system should be able to add functions flexibly.*

*5.1.3 Resource usage rate:*

*The PLVS CPU usage <=50%.*

*Memory usage <=50%.*

*5.1.4 Business volume:*

*The PLVS can simultaneously satisfy 10,000 user requests and provide browsing capabilities for 25,000 concurrent users.*

*5.1.5 System capacity:*

*The number of database table rows does not exceed 1 million rows, the maximum capacity of the database does not exceed 1000 GB, and the disk space needs at least 40G.*

*5.1.6 System specific requirement:*

* + - * + The system read and write accesses on the user's computer.
        + There can not be any errors in the input file (ie. bad formatting, more than one vote per ballot).
        + Input files must contain at least one ballot
        + 100,000 ballots need to be processed in less than 5 minutes.
        + Ensuring one vote for one person is handled at the voting centers.

## Safety Requirements

*The PLVS system should be able to handle various abnormal situations that occur during system operation, such as abasing personal error, the input of illegal data, failure of hardware devices, etc.*

*If the PLVS failed to open vote files, this should be recorded in the log and send to the management center. The system should be able to handle it properly and try to avoid these problems. Therefore, the system should be checked and updated regularly to ensure that the input is prompted, the data is checked, and the data is prevented from being abnormal. Ensure that the probability of failing to complete the business due to the failure of the software system is less than 5‰.*

## Security Requirements

*The PLVS should have strict access control. Security such as ensuring one vote for one person is handled at the voting centers. For example, only the government officials and the operators can assess the system, if anyone else wants to log into the system should be denied. After the user is authenticated, the user can only access the data within the scope of his authority, it can only perform operations within its scope of authority. Different users have different identities and rights. But none of the users can edit the results of voting, they can only view the results and do some summary. They need to provide trusted authorization management services under the premise of the authenticity of users, to protect data from illegal or unauthorized access and tampering, and to ensure the confidentiality and integrity of data. For example, the interface permissions of the government officials and operators are different, and the operators should not have the same access rights as the government officials. In addition, the system should provide operational log management and security auditing capabilities to track the historical usage of the system.*

## Software Quality Attributes

## *The PLVS guarantees the availability of the system under conditions of the poor network environment. And provide data backup and recovery functions, so that when the system data is lost due to system errors or other reasons or the system data is destroyed, the data can be restored and restored in time (this function is provided by hardware and third-party software).*

## Business Rules

Need to abide by certain laws and regulations, such as sometimes need to keep the audit log and other information according to the law. For the PLVS, voting information cannot be stolen by outsiders, nor can it be used for any type of commercial activity to copy, sell or rent voting results.

# Other Requirements

## Randomization:

In the event of a tie, the voting system software will use random number generation to decide the winner. As computer driven randomization is not truly random, and to ensure that these tie-breaks are fair the randomization will be seeded and ran several times at the software's launch in order to avoid these potential issues as much as possible.

Appendix A: Glossary

1. Voter - A member of the electorate casting a ballot in an election.
2. Candidate - A member of a political party running for office
3. Political Party (Party) - An organization of candidates and government officials sharing common ideals, and working together to actualize them.
4. Ballot - The record of an individual vote.
5. Tie - The event where multiple different candidates or parties have the same number of votes.
6. CSELabs - The University of Minnesota's Science of College and Engineering computer lab machines, as well as what can be emulated on https://vole.cse.umn.edu.
7. GB - Acronym representing Gigabyte
8. PLVS - Acronym representing Party List Voting System
9. Party-List Voting (PLV) - A voting algorithm allowing political parties to select a candidate for each electoral district, and allows each voter to vote for a party. Political seats are then assigned based the proportion of the vote each party receives and the order of each party's candidates.
   1. Closed-Party-List Voting (CPLV) - A variant of the PLV algorithm in which parties decide the order of their candidates prior to election, and the voters vote for a party.
   2. Open-Party-List Voting (OPLV) - A variant of the PLV algorithm in which the order of a party's candidates is determined by the voters, who vote for a single candidate. This vote also counts towards that candidate's party.

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>