

---

# **Software Requirements Specification**

for

## **Voting System**

**Version 1.0 approved**

**Prepared by Ge Yu, Songyan Wu, Hongyu Zhu**

**University of Minnesota CSCI5801**

**10/05/2019**

# Table of Contents

<b>0. Revision History</b>	<b>3</b>
<b>1. Introduction</b>	<b>4</b>
1.1 Purpose	4
1.2 Document Conventions	4
1.3 Intended Audience and Reading Suggestions	4
1.4 Product Scope	4
1.5 References	5
<b>2. Overall Description</b>	<b>5</b>
2.1 Product Perspective	5
2.2 Product Functions	5
2.3 User Classes and Characteristics	6
2.4 Operating Environment	6
2.5 Design and Implementation Constraints	6
2.6 User Documentation	7
2.7 Assumptions and Dependencies	7
<b>3. External Interface Requirements</b>	<b>7</b>
3.1 User Interfaces	7
3.2 Hardware Interfaces	9
3.3 Software Interfaces	9
3.4 Communications Interfaces	10
<b>4. System Features</b>	<b>10</b>
4.1 Prompt and Read	10
4.2 Produce audit file	13
4.3 Show election progress	17
4.4 Display winner and information of the election	18
<b>5. Other Nonfunctional Requirements</b>	<b>22</b>
5.1 Performance Requirements	22
5.2 Safety Requirements	22
5.3 Security Requirements	22
5.4 Software Quality Attributes	22
5.5 Business Rules	23
<b>6. Other Requirements</b>	<b>23</b>

## 0. Revision History

Name	Date	Reason For Changes	Version
G. Yu, H. Zhu, S. Wu	10/05	Original Documentation, parts are not done yet	1.0.0
G. Yu, H. Zhu, S. Wu	10/07	Use case and function requirements inserted	1.0.1
G. Yu, H. Zhu, S. Wu	10/10	Product functions updated, format fixed	1.0.2
G. Yu, H. Zhu, S. Wu	10/21	Use-case changed	1.0.3

# **1. Introduction**

## **1.1 Purpose**

The purpose of this documentation is to provide a description of Voting system. It explains the purpose and features of the systems, along with the interfaces of the system, what the system will do, the constraints under which it must operate, and how different classes interact with others. This document is intended for both developers and software users.

## **1.2 Document Conventions**

This document was created based on the IEEE template for System Requirement Specification Documents.

## **1.3 Intended Audience and Reading Suggestions**

- Voters, who are involved to express their preference for candidates or parties in such election process
- Candidates, who are nominated in the party lists of the election with their parties
- Party members, who belong to the parties that are in the election
- General public, who is awe of such election and want to acknowledge the out come of this election

## **1.4 Product Scope**

Voting system is a tool that implements the ability to perform two types of voting: open party list voting and closed party list voting. Open party list voting allows voters to express a preference for particular candidates, not just parties. It is designed to give voters some say over the order of the list and thus which candidates get elected. In a closed list system -- the original form of party list voting -- the party fixes the order in which the candidates are listed and elected, and the voter simply casts a vote for the party as a whole. This voting system is designed to replicate the election

process and display the outcomes. The intended audience which stated above will be able to learn about the election through this voting system.

## **1.5 References**

Voting System requirement:

<https://canvas.umn.edu/courses/134519/files/8682408/download?wrap=1>

## **2. Overall Description**

### **2.1 Product Perspective**

Voting System is developed for everyone that would like to know the voting process and results, such as the voters, the analytical staff or even the party representatives. It has some functions like prompt, read the file, and also have options to choose based on different Open Party List or Closed Party List. With this software, the process and result can be displayed directly through the screen, guarantees fairness and efficiency. This software can run on different platforms such as Windows, Mac Os and Linux.

### **2.2 Product Functions**

For details, refer to section 3.1. User Interfaces

System:

- Read: System can read the voting result based on the text file.
- Prompt: System is able to prompt the user with filename.
- Audit File: System is able to produce an audit file for future analyzing based on the voting type
- Display result: System can display the election result
- Display voting process: System is able to display the process of voting

Voter:

- Run: User can run the software

## **2.3 User Classes and Characteristics**

- System: Since system is the main actor in this software, most operation will be finished automatically by the system.
- Audiences: Audiences who would like to see the results.

## **2.4 Operating Environment**

- Windows 2000
- Windows XP
- Windows Vista
- Windows 7
- Windows 8
- Windows 10
- Mac OS X
- Linux
- Ubuntu 16
- Ubuntu 18

## **2.5 Design and Implementation Constraints**

Voting System is developed in C++ and uses Visual Studio for its visualization interaction. It uses a modular design where every feature is wrapped into a separate module, and the modules depend on each other through well-written APIs. There are several APIs available to make plugin development easy.

## 2.6 User Documentation

There will be a short documentation basically introducing the software's results' UI, also how to run the program, how does each step look to give the user an overall sense of the software.

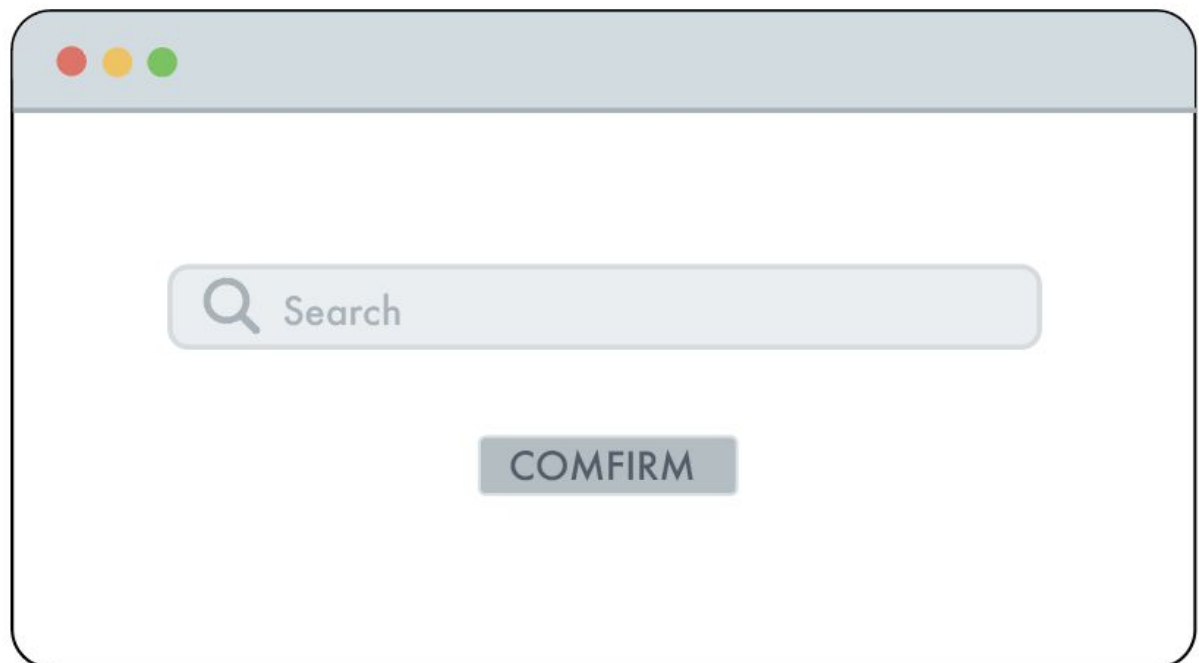
## 2.7 Assumptions and Dependencies

Voting System is developed in C++ and therefore requires C++ and xcode to be installed on the user's system. The stable version of C++ requires the latest version. This applies to Windows, Linux users, and MacOS users.

# 3. External Interface Requirements

## 3.1 User Interfaces

1. Prompt filename



## 2. Generated audit file

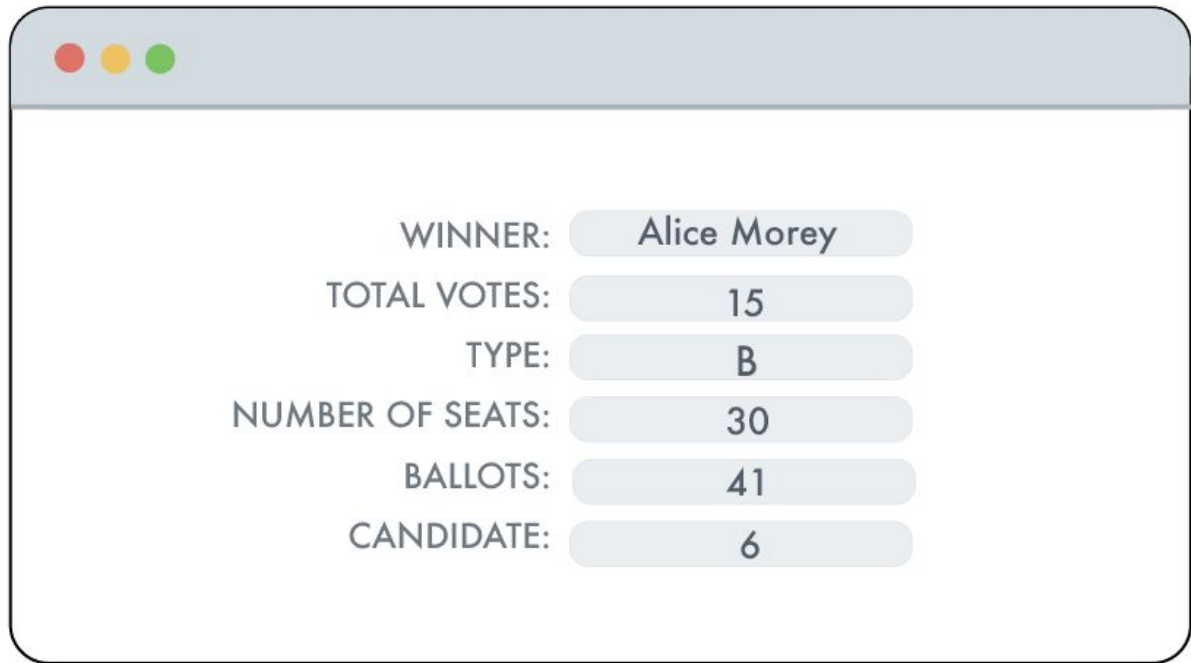


## 3. Show election progress

A:15	B:16	C:10
Ben Foster: 10	Steven Wong: 1	Sarah Pingle: 5
Wendy Berg: 5	Alice Morey: 15	Colin Volz: 5



#### 4. Display winner



### 3.2 Hardware Interfaces

The minimum hardware requirements of Voting system is the same for C++ which requires 1 GB RAM, 9-58 GB free hard disk space depending on edition and configuration, including space required for temporary files, DVD-ROM drive (if installing from a Media Kit DVD), and basic GPU – Any vendor DirectX 9.0 class or better.

### 3.3 Software Interfaces

Voting system is developed in C++ and therefore requires C++ and xcode to be installed on the user's system. Additional information can be found in section 2.7 of this document.

### 3.4 Communications Interfaces

Voting system requires user to provide with their excel file about the voting results, user is able to decide their way of providing files based on their preferences. If they would like to download the file online, they might need Internet connection to download the file, or they can use a USB drive. However, the software itself doesn't need Internet connection to run, in other words, no other communications is required.

## 4. System Features

This section demonstrates Voting System some of the most prominent features and explains how the system works to prompt, read file, display and show the results.

### 4.1 Prompt and Read

#### 4.1.1 Description and Priority

The system will prompt users with the file name that will be read, then automatically read the file stored in the local disk, then generate a file with corresponding format.

Since the benefit of this feature is based on the file that will be read stored on the disk, the priority of this is 10, which is the most important, since if there is no file, the whole system won't be running.

#### 4.1.2 Stimulus/Response Sequences

User: Click the icon which represents the voting system

System: Open the software

User: Click the the button "Run"

System: System prompt the user to enter the file name

User: Enter the file name that will be used to run this software

System: Select the file with the name to open it

System: Check if the file with the entered name exist

System: Check if the file is openable

System: Start reading

System: Determine the type of the file is open/closed party list

System: Generate a corresponding file with correct format

#### 4.1.3 Functional Requirements

REQ-1:

Name	Prompt User for filename
ID	user_001
Description	The user will be prompted for entering filename
Actor	User, System
Organizational Benefits	Users will know which file is used to display the result.
Frequency of Use	Whenever the user runs the system
Triggers	The user selects to run the system
Preconditions	<ul style="list-style-type: none"> <li>• A text file with ballots results is provided</li> <li>• The text file is openable</li> <li>• The file is in the same directory as the program</li> <li>• File is given the determined format</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>• User successfully enter the name</li> <li>• File with the entered name is ready to open</li> </ul>
Main Course	<ol style="list-style-type: none"> <li>1. User open the program</li> <li>2. User decides to run the program</li> <li>3. System prompts user the filename</li> <li>4. User enter the filename (See AC1) (See EX1)</li> <li>5. System prompt user successfully found the file</li> </ol>
Alternate Courses	AC1: User enter the wrong file name <ol style="list-style-type: none"> <li>1. Prompt user with information no file with that name is found.</li> <li>2. Redirect to "User enter the file name"</li> </ol>
Exceptions	EX1: file can not be opened <ol style="list-style-type: none"> <li>1. The program will display an error message about file is</li> </ol>

	not able to be opened. 2. Redirect to "User enter the file name"
--	---

## REQ-2:

Name	Read file
ID	System_002
Description	The <b>system</b> need to read in the file
Actor	<b>System</b>
Organizational Benefits	<ul style="list-style-type: none"> <li>It ensure the correctness of the voting system</li> <li>It saves user's time and work to read files individually</li> </ul>
Frequency of Use	Whenever user selects a file to read
Triggers	The user select the option to read file
Preconditions	<ul style="list-style-type: none"> <li>File is not empty</li> <li>File has correct format for system to read</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>User is able to read a file</li> <li>File is exported from Excel to into the CSV format</li> </ul>
Main Course	<ol style="list-style-type: none"> <li>System prompt user to read file</li> <li>User input filename and select to read file</li> <li>System determines if the file is empty (see EX1)</li> <li>File is not empty</li> <li>System determines if the first line of the file is open party list or closed party list (see AC1, AC2) (see EX2)</li> <li>System export file from Excel into CSV format</li> </ol>
Alternate Courses	<p>AC1: First line of the file is "OPL"</p> <ol style="list-style-type: none"> <li>Direct user to user case "Produce audit file for Open Party List"</li> </ol> <p>AC2: First line of the file is "CPL"</p> <ol style="list-style-type: none"> <li>Direct user to user case "Produce audit file for Closed Party List"</li> </ol>
Exceptions	<p>Ex1: No value display on open party list</p> <ol style="list-style-type: none"> <li>User cannot read file due to missing data</li> </ol> <p>Ex2: First line of the file is not OPL or CPL</p> <ol style="list-style-type: none"> <li>File does not belong to open party list voting nor closed</li> </ol>

	party list voting
--	-------------------

## 4.2 Produce audit files

### 4.2.1 Description and Priority

The system will produce audit files after reading the file. For the open party list voting, the information which includes the winner, type of election, number of seats, number of ballots, number of candidates will be generated into audit file. For the closed party list voting, the information which includes type of election, number of parties, parties in order of ballot ordering, number of seats, number of ballots, number of candidates will be generated into audit file. The priority for such a function is also 10. Because audit files are required documents to continue next steps. It ensures the correctness and fairness of the election .

### 4.2.2 Stimulus/Response Sequences

System: determines if the file is Excel format

System: determines if the first line of the file is open party list

System: determines if the second line of the file is number of seats

System: determines if the third line of the file is number of ballots

System: determines if the third line of the file is number of candidates

System: checks the format the inputs for candidates in [ ]

System: counts the votes

System: Audit file is produced successfully

System: Return to Read file user case to export file into CSV file

System: determines if the first line of the file is closed party list

System: determines if the second line of the file is number of parties

System: determines if the third line of the Parties in Order of Ballot Ordering

System: determines if the forth line of the file is number of seats

System: determines if the fifth line of the file is number of ballots

System: determines if the fifth line of the file is number of candidates

System: counts the votes

System: Audit file is produced successfully

#### 4.2.3 Functional Requirements

Use-case 3:

Name	Produce audit file for Closed Party List
ID	aduit_003
Description	The system will produce audit file with the election information at the time
Actor	System
Organizational Benefits	<ul style="list-style-type: none"> <li>It ensures the correctness and fairness of the voting system</li> </ul>
Frequency of Use	Whenever read a file
Triggers	The file is determined to be open party list file from read file step
Preconditions	<ul style="list-style-type: none"> <li>The file in read file user case is determined to be open party list file</li> <li>The file is Excel format</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>Audit file is successfully produced</li> <li>No duplicated audit file with same filename is produced</li> </ul>
Main Course	<ol style="list-style-type: none"> <li>System determines if the file is Excel format (see EX1)</li> <li>File is Excel format</li> <li>System determines if the first line of the file is closed party list (see EX2)</li> <li>Type of election is closed party list</li> <li>System determines if the second line of the file is number of parties (see EX3)</li> <li>Number of parties are valid</li> <li>System determines if the third line of the Parties in Order of Ballot Ordering (see EX3)</li> <li>Parties in Order of Ballot Ordering are valid</li> <li>System determines if the forth line of the file is number of seats (see EX3)</li> <li>Number of seats are valid</li> <li>System determines if the fifth line of the file is number of ballots(see EX3)</li> <li>Number of seats are ballots</li> </ol>

	13. System determines if the fifth line of the file is number of candidates (see EX3) 14. Formats are correct 15. System counts the votes(see EX4) 16. Audit file is produced successfully 17. Return to Read file user case step 6
Alternate Courses	AC1: System shows error on producing audit file 1. Return to Read file user case
Exceptions	Ex1: File is not in Excel format 1. File is not in Excel format to read  Ex2: First line of the file is not CPL 1. File does not belong to open party list voting 2. Return to User case "Read file"  EX3: Type is incorrect 1. Invalid input  Ex4: Input format is not [char,char,char] 1. Incorrect format in [ ] for candidate

## REQ-4:

Name	Produce audit file for Open Party List
ID	aduit_004
Description	The system will produce audit file with the election information at the time
Actor	System
Organizational Benefits	<ul style="list-style-type: none"> <li>It ensures the correctness and fairness of the voting system</li> </ul>
Frequency of Use	Whenever read a file
Triggers	The file is determined to be open party list file from read file step
Preconditions	<ul style="list-style-type: none"> <li>The file in read file user case is determined to be open party list file</li> <li>The file is Excel format</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>Audit file is successfully produced</li> </ul>

	<ul style="list-style-type: none"> <li>No duplicated audit file with same filename is produced</li> </ul>
Main Course	<ol style="list-style-type: none"> <li>System determines if the file is Excel format (see EX1)</li> <li>File is Excel format</li> <li>System determines if the first line of the file is open party list (see EX2)</li> <li>Type of election is open party list</li> <li>System determines if the second line of the file is number of seats (see EX3)</li> <li>Number of seats are valid</li> <li>System determines if the third line of the file is number of ballots (see EX3)</li> <li>Number of ballots are valid</li> <li>System determines if the third line of the file is number of candidates (see EX3)</li> <li>Number of candidates are valid</li> <li>System checks the format the inputs for candidates in [ ] (see EX4)</li> <li>Formats are correct</li> <li>System counts the votes(see EX5)</li> <li>Audit file is produced successfully</li> <li>Return to Read file user case step 6</li> </ol>
Alternate Courses	AC1: System shows error on producing audit file <ol style="list-style-type: none"> <li>Return to Read file user case</li> </ol>
Exceptions	<p>Ex1: File is not in Excel format</p> <ol style="list-style-type: none"> <li>File is not in Excel format to read</li> </ol> <p>Ex2: First line of the file is not OPL</p> <ol style="list-style-type: none"> <li>File does not belong to open party list voting</li> <li>Return to User case "Read file"</li> </ol> <p>Ex3: Input is not a number</p> <ol style="list-style-type: none"> <li>Invalid input for number of seats</li> </ol> <p>Ex4: Input format is not [char,char]</p> <ol style="list-style-type: none"> <li>Incorrect format in [ ] for candidate</li> </ol>



## 4.3 Show election progress

### 4.3.1 Description and Priority

The screen will show how the election is progressing, all parties and candidates will be listed, and each vote will be directly added to the candidate's total votes. The priority for election progress should be highest, it guarantees the correctness of the election, which means it should have a scale of 10.

### 4.3.2 Stimulus/Response Sequences

User: Click to view the election progress

System: Determine the type of the election

System: Open the correct audit file according to the type of the election

System: System display a option to share with median

User: Choose to share with median or not

### 4.3.3 Functional Requirements

REQ-5:

Name	Show election progress
ID	progress_007
Description	The audit file can be read and shared with media personals
Actor	Programmers, testers, and election officials
Organizational Benefits	<ul style="list-style-type: none"><li>• It visualize the process of voting, which makes it clearer</li><li>• Audit file could replicate election itself</li></ul>
Frequency of Use	Whenever user wants to view the election progress
Triggers	The user wants to view the election progress
Preconditions	<ul style="list-style-type: none"><li>• The audit files are generated</li></ul>
Postconditions	<ul style="list-style-type: none"><li>• The voting process is replicated</li></ul>

	<ul style="list-style-type: none"> <li>The user is able to share with the media personnels</li> </ul>
Main Course	<ol style="list-style-type: none"> <li>System open the correct audit file(see AC1)(see AC2)(see EX1)</li> <li>The file is displayed in the correct ballot format</li> <li>System displays the option to share</li> <li>User select media personnels to share with</li> </ol>
Alternate Courses	<p>AC1: Type of election is OPL</p> <ol style="list-style-type: none"> <li>System opens the OPL audit file</li> <li>Return to main course step 2</li> </ol> <p>AC2: Type of election is CPL</p> <ol style="list-style-type: none"> <li>System opens the CPL audit file</li> <li>Return to main course step 2</li> </ol>
Exceptions	<p>Ex1: Audit file input is missing</p> <ol style="list-style-type: none"> <li>Cannot open audit file due to missing data</li> </ol>

## 4.4 Display winner and information of the election

### 4.4.1 Description and Priority

The results of the election and the winner will be displayed on the screen. For the open party list voting, the information which includes the winner, type of election, number of seats, number of ballots, number of candidates will be displayed. For the closed party list voting, the information which includes type of election, number of parties, parties in order of ballot ordering, number of seats, number of ballots, number of candidates will be displayed. The priority for such a function is also 10, since the result is important to display to the screen. It ensures the correctness of the election results are available to its audiences.

### 4.4.2 Stimulus/Response Sequences

System: Determines what type of election(CPL/OPL)

Type of election is CPL

System: Determines if the highest votes

System: Winner is determined and winner's party is displayed on screen

System: Type of election is displayed

System: Number of Parties is displayed

System: Parties in Order of Ballot Ordering is displayed

System: Number of seats is displayed

System: Number of Ballots is displayed

System: Number of Candidates is displayed

Type of election is OPL

System: Determines the highest votes

System: Winner is determined and displayed on screen

System: Type of election is displayed

System: Number of seats is displayed

System: Number of ballots is displayed

System: Number of candidate is displayed

#### 4.4.3 Functional Requirements

##### REQ 6:

Name	Display for closed party list
ID	screen_005
Description	The screen will display the winner and information about the election
Actor	System
Organizational Benefits	<ul style="list-style-type: none"> <li>• It gives a direct view of election results</li> <li>• It helps people have better understanding</li> </ul>
Frequency of Use	Whenever program is done
Triggers	System finish counting ballots
Preconditions	<ul style="list-style-type: none"> <li>• System has finished counting ballots</li> </ul>
Postconditions	<ul style="list-style-type: none"> <li>• The results is displayed to screen</li> </ul>

	<ul style="list-style-type: none"> <li>The information does not contain a long audit of the votes</li> </ul>
Main Course	<ol style="list-style-type: none"> <li>System determines what type of election(see AC1)(see EX1)</li> <li>Type of election is CPL</li> <li>System determines if the highest votes (see AC2)</li> <li>Winner is determined and winner's party is displayed on screen</li> <li>Type of election is displayed</li> <li>Number of Parties is displayed</li> <li>Parties in Order of Ballot Ordering is displayed</li> <li>Number of seats is displayed</li> <li>Number of Ballots is displayed</li> <li>Number of Candidates is displayed</li> </ol>
Alternate Courses	<p>AC1: The type of election is OPL</p> <ol style="list-style-type: none"> <li>Display for open party list</li> </ol> <p>AC2: There is tie on votes</p> <ol style="list-style-type: none"> <li>Randomly select the winner in fair coin toss</li> <li>Return to Main course step 3</li> </ol>
Exceptions	<p>Ex1: First line of the file is not CPL or OPL</p> <ol style="list-style-type: none"> <li>File does not belong to open party list voting nor closed party list voting</li> </ol>

## REQ 7:

Name	Display for open party list
ID	screen_006
Description	The screen will display the winner and information about the election
Actor	System
Organizational Benefits	<ul style="list-style-type: none"> <li>It gives a direct view of election results</li> <li>It helps people have better understanding</li> </ul>
Frequency of Use	Whenever program is done
Triggers	System finish counting ballots

Preconditions	<ul style="list-style-type: none"><li>• System has finished counting ballots</li></ul>
Postconditions	<ul style="list-style-type: none"><li>• The results is displayed to screen</li><li>• The information does not contain a long audit of the votes</li></ul>
Main Course	<ol style="list-style-type: none"><li>1. System determines what type of election (see AC1)(see EX1)</li><li>2. Type of election is OPL</li><li>3. System determines the highest votes (see AC2)</li><li>4. Winner is determined and displayed on screen</li><li>5. Type of election is displayed</li><li>6. Number of seats is displayed</li><li>7. Number of ballots is displayed</li><li>8. Number of candidate is displayed</li></ol>
Alternate Courses	<p>AC1: Type of election is CPL</p> <ol style="list-style-type: none"><li>1. Return to display for closed party list step 1</li></ol> <p>AC2: There is tie on votes</p> <ol style="list-style-type: none"><li>1. Randomly select the winner in fair coin toss</li><li>2. Return to Main course step 3</li></ol>
Exceptions	<p>Ex1: First line of the file is not OPL or CPL</p> <ol style="list-style-type: none"><li>1. File does not belong to open party list voting nor closed party list voting</li></ol>

## **5. Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

The minimum hardware requirements of Voting system is same for C++ which requires 1 GB RAM, 9-58 GB free hard disk space depending on edition and configuration, including space required for temporary files, DVD-ROM drive (if installing from a Media Kit DVD), and basic GPU – Any vendor DirectX 9.0 class or better.

### **5.2 Safety Requirements**

For this voting system, there is no special safety requirement. All the files will be given in a determined format.

### **5.3 Security Requirements**

For this voting system, 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**

Voting system provides the users with both simple and advanced features. Due to its well designed and easy to use interface it can be used by both experts and typical users. All the instructions are clear enough for the users to read and understand thus no basic knowledge needed before using it.

## 5.5 Business Rules

Voting system provides the same authorization to the general public. Common users, such as voters, candidates, party members and the general public will use this system to replicate the election process and view the results.

## 6. Other Requirements

### Appendix A: Glossary

- UI: User interface
- API: Application programming interface
- Voters: who are involved to express their preference for candidates or parties in such election process
- Candidates: who are nominated in the party lists of the election with their parties
- Party members: who belong to the parties that are in the election
- General public: who is aware of such election and want to acknowledge the outcome of this election
- OPL: Open party list
- CPL: Closed party list
- EOF: End of file
- Voting Process: How the votes are distributed
- Display: Print a diagram/text as a result to the screen

## **Appendix B: Analysis Models**

See Appendix C.2

## **Appendix C: To Be Determined List**

1. The user interface needs to be determine and update in the future
2. The analysis models will be generated after we update the user interface, and also after we decide the overall algorithm
3. User's manual needs to be documented
4. The details for hardware interface requirements will be determined
5. The certain algorithm which has a higher efficiency needs to be determined