
Software Requirements Specification

for

Voting System(VS)

Version 1.0 approved

Prepared by Junren Huang, Jian Wang, Pengyin Chen, Shuai Hao

University Of Minnesota at Twin Cities

February 20, 2020

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	4
2.4 Operating Environment	5
2.5 Design and Implementation Constraints	5
2.6 User Documentation	5
2.7 Assumptions and Dependencies	5
3. External Interface Requirements	6
3.1 User Interfaces	6
3.2 Hardware Interfaces	6
3.3 Software Interfaces	6
3.4 Communications Interfaces	7
4. System Features	7
4.1 Voting Information Set-Up	7
4.2 Shuffle/Shuffle Off	8
4.3 System Generates The Audit Report	8
4.4 Help window	9
4.5 Testing	9
5. Other Nonfunctional Requirements	10
5.1 Performance Requirements	10
5.2 Safety Requirements	10
5.3 Security Requirements	10
5.4 Software Quality Attributes	10
5.5 Business Rules	10
6. Other Requirements	10
Appendix A: Glossary	11
Appendix B: Analysis Models	11
Appendix C: To Be Determined List	11

Revision History

Name	Date	Reason For Changes	Version
VS	02/20/20	First time created	1.0
VS	02/21/20	Detail modification	1.0

1. Introduction

1.1 Purpose

The purpose of this document is to establish the description of the Voting system. This document will explain the essential purpose and features of the system. This document will explain the primary purpose and features of the system. It also includes the interface of the system, the primary function of the system. And the constraint of the system and how the system responds to external stimuli. This document not only intends to regular users but also for developers.

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

- General users like voting election officials who are involved in the process of campaigns.
- Advanced/Professional users, such as engineers or researchers, who want to use the voting system for more demanding class activities.
- Programmers who are interested in working on the project by further developing it or fixing existing bugs.

1.4 Product Scope

Voting System is a tool that users can use to input information, run desired election algorithms with shuffle option, and produce audit reports. Users can use it to manage voting

information and interact with them. They can produce reports with statistical information of each candidate, testing and so on so that it could improve management efficiency and achieve fairness of the campaign process.

1.5 References

IEEE Template for System Requirement Specification Documents:

<https://goo.gl/nsUFwy>

Voting System's GitHub page:

<https://github.umn.edu/umn-csci-5801-002-s20/repo-Team5>

2. Overall Description

2.1 Product Perspective

The voting system was developed to ensure the fairness and correctness of the voting count. Mainly the voting system is separate from the actual voting. Therefore, all voting data is collected before the system runs. The only function of this system is to count the candidate by users' requirement.

It is a team project, and it has a very active developer team to support it and provide feedback to users. It was developed to run on Windows, Mac OS X, and Linux.

2.2 Product Functions

Mainly this section describes the process of the system works. Due to the ballot collection being separate from this system. Therefore, this system is not responsible for the data collection. The system is only responsible for data counting.

The process of the system is shown below:

1. Firstly, the system prompts the user type in how many seats there are to fill and the algorithm to use, the user can select which data they want to use.
2. The system reads ballot data from existing files to gather test data.
3. System runs the election with the algorithm that the user has chosen with the user's required number of seats.
4. System determines the number of candidates and the number of ballots using the files themselves.
5. After the election has been run, the system displays the details of the election to the screen, such as the number of ballots, the number of seats, and the number of candidates. Also, the system generates an audit report.

2.3 User Classes and Characteristics

The user of the voting system is expected to be internet literate and familiar with computer software and be able to use buttons, menus, scroll bar and similar tools.

2.4 Operating Environment

- Windows 7/8/10
- OS X
- Linux

2.5 Design and Implementation Constraints

The Voting System is developed by Java. It uses a modular design where every feature is wrapped into a separate module and the modules depend on each other through well.

2.6 User Documentation

There will be a user guide.

2.7 Assumptions and Dependencies

VS is developed in Java and therefore requires Java to be installed on the user's system.

The latest stable version of the voting system requires Java version 7 or higher. This applies to Windows and Linux users. On Mac OS X, Java is bundled with the application.

3. External Interface Requirements

3.1 User Interfaces

There will be interfaces for general users to prompt input information, generate result reports, and the help window.

3.2 Hardware Interfaces

The minimum hardware requirements of the voting system are a 500 Megahertz CPU and 128 megabytes of RAM. Also, a compatible graphics card is required to make sure the voting system could have high-quality representation.

3.3 Software Interfaces

VS requires Java to be installed on the system, more specifically Java version 7 or 8 for its latest release. There is no need to have a database since all input information was stored in a file.

3.4 Communications Interfaces

All processes of VS are done locally, thus we do not need a communication interface with the external source.

4. System Features

This section demonstrates the VS's most prominent features and explains how they can be used and the results they will give back to the user.

4.1 Voting Information Set-Up (Use Case-UC001/002)

4.1.1 Description and Priority

System allows users to provide voting information. Users can input filename, seats, and choose algorithms. This is a high priority feature, many features depend on this feature.

4.1.2 Stimulus/Response Sequences

1. System prompts users to input filenames/seats/ choose algorithms
2. User adds information as they desire
3. System stores the input information
4. System reads the filename, extract information from the file and store it.

4.1.3 Functional Requirements

None.

4.2 Shuffle/Shuffle off (Use Case-UC003)

4.2.1 Description and Priority

System will give users the option to shuffle balloons. This feature makes the system more flexible to have the capability of testing and ensure fairness and reliability in campaigns. This is a high priority feature, and this is required for some other features.

4.2.2 Stimulus/Response Sequences

1. System prompt user to choose shuffle or not shuffle

2. User choose their desired option
3. System verifies their choice
4. System redirects users to the welcome page

4.2.3 Functional Requirements
None

4.3 System Generates The Audit Report (Use Case-UC004)

4.3.1 Description and Priority

After the system gets enough information, it could start running to get the result and generate an audit report which includes statistics and information of each candidate. This feature improves management efficiency and simplifies the processes of campaigns. Also, it gets organized statistics and improves data readability for users. This is a low priority feature.

4.3.2 Stimulus/Response Sequences

1. System running to get results by using available input information
2. System displays to the screen providing the details of the election when it is running
3. After it is done, the system generates audit report and show the location of it

4.3.3 Functional Requirements

- REQ-1: Voting Information Set-Up
REQ-2: Shuffle/Shuffle off

4.4 Help window (Use Case-UC005)

4.4.1 Description and Priority

A help window is provided to users with information on how to run this program. This feature helps users eliminate confusion and get some useful knowledge of how to interact with this program. This is a low priority feature.

4.4.2 Stimulus/Response Sequences

1. User requests to open the help window
2. System opens the window and shows useful information to users

4.4.3 Functional Requirements

None

4.5 Testing (Use Case-UC006)

4.5.1 Description and Priority

Before we actually run our system, we need to test it to make sure its functionality. By using the shuffle off option, we could run some test files and check the results with expected results to finish testing. This feather helps us to figure out existing or potential problems within our system and improve the robustness of our system. This feather has a low priority.

4.4.2 Stimulus/Response Sequences

1. User goes into testing mode by running program with command line arguments
2. User inputs testing file and necessary information
3. System generates audit file
4. System compares the result with expected result
5. System output testing results.

4.4.3 Functional Requirements

- REQ-1: Voting Information Set-Up
- REQ-2: Shuffle/Shuffle Off
- REQ-3: System Generates The Audit File

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The voting system requires a system with at least a 500 megahertz CPU and 128 megabytes of RAM. Performance depends on the size and as a result, the system requirements for higher quality are more demanding.

5.2 Safety Requirements

There are no safety requirements for the Voting system.

5.3 Security Requirements

Voting System has no security issues to be handled at this point. The security of the voting will be handled at the voting precinct itself. The files you will receive will have been transmitted to you via a secure means.

5.4 Software Quality Attributes

Voting System provides the user 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. However, users must already have a basic knowledge of computers before using it.

5.5 Business Rules

None

6. Other Requirements

None

Appendix A: Glossary

References: https://en.wikipedia.org/wiki/Main_Page

- Shuffle: Shuffling is a procedure used to randomize a deck of playing cards to provide an element of chance in card games. Shuffling is often followed by a cut, to help ensure that the shuffler has not manipulated the outcome..
- Audit: An audit is a systematic and independent examination of books, accounts, statutory records, documents and vouchers of an organization to ascertain how far the financial statements as well as non-financial disclosures present a true and fair view of the concern.

Appendix B: Analysis Models

The analysis model will be based on the prerequisite settings read from input information like seats number, algorithm to use, shuffle option, and input file. Our system will run results, display data and generate audit reports. We expand functionality that they may use in different situations.

Appendix C: To Be Determined List

- Welcome page
- user guide
- User interface design
- Expected exception message