
Software Requirements Specification

for

Project 1 - Waterfall Methodology

Version 1.1 approved

**Prepared by Zachary Tressel
Isabel Tomb
Ian Luck
Jessica Moore**

Team 10

2/16/20

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	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	3
2.5 Design and Implementation Constraints	3
2.6 User Documentation	3
2.7 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	4
3.3 Software Interfaces	4
3.4 Communications Interfaces	4
4. System Features	4
4.1 Input Seats to Fill	4
4.2 Input Voting Algorithm	5
4.3 Input File Location	5
4.4 Help Menu	6
4.5 Run Election	7
4.6 View Election Results	8
4.7 Create Log File	9
4.8 Turn Shuffle Option Off	10
5. Other Nonfunctional Requirements	10
5.1 Performance Requirements	10
5.2 Safety Requirements	11
5.3 Security Requirements	11
5.4 Software Quality Attributes	11
5.5 Business Rules	11
6. Other Requirements	11
Appendix A: Glossary	11
Appendix B: Analysis Models	12
Appendix C: To Be Determined List	12

Revision History

Name	Date	Reason For Changes	Version
Initial Draft	2/16/20	Starting project 1 SRS	1.0
Final Draft	2/21/20	Finishing project 1 SRS	1.1

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Voting System software. It will explain the features of the software, the interfaces of the software, what the software will do, and the constraints under which it must operate. This document is intended for election officials who will use the software. The final version of this document is version 1.1.

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

This document is for developers who want to understand how this software works. This document can be read in any order, including start to finish. The document is divided in sections based off of the provided template and includes many relevant use cases.

1.4 Product Scope

The software described in this document is for a voting system. The software will allow the user to run two different types of voting systems: a plurality system and a single transferable voting (STV) system using a droop quota. The user will input how many seats there are to fill and the algorithm to use at the beginning of the program. The user can also run a test file and turn the shuffle option off. After the software has been run, the results of the election will be displayed on the screen.

1.5 References

Team 10's GitHub repository:

<https://github.umn.edu/umn-csci-5801-002-s20/repo-Team10>

IEEE Template for System Requirement Specification Documents:

<https://goo.gl/nsUFwy>

Software Requirements Specification (SRS) Document Directions

<https://canvas.umn.edu/courses/158173/files/11094361/download?wrap=1>

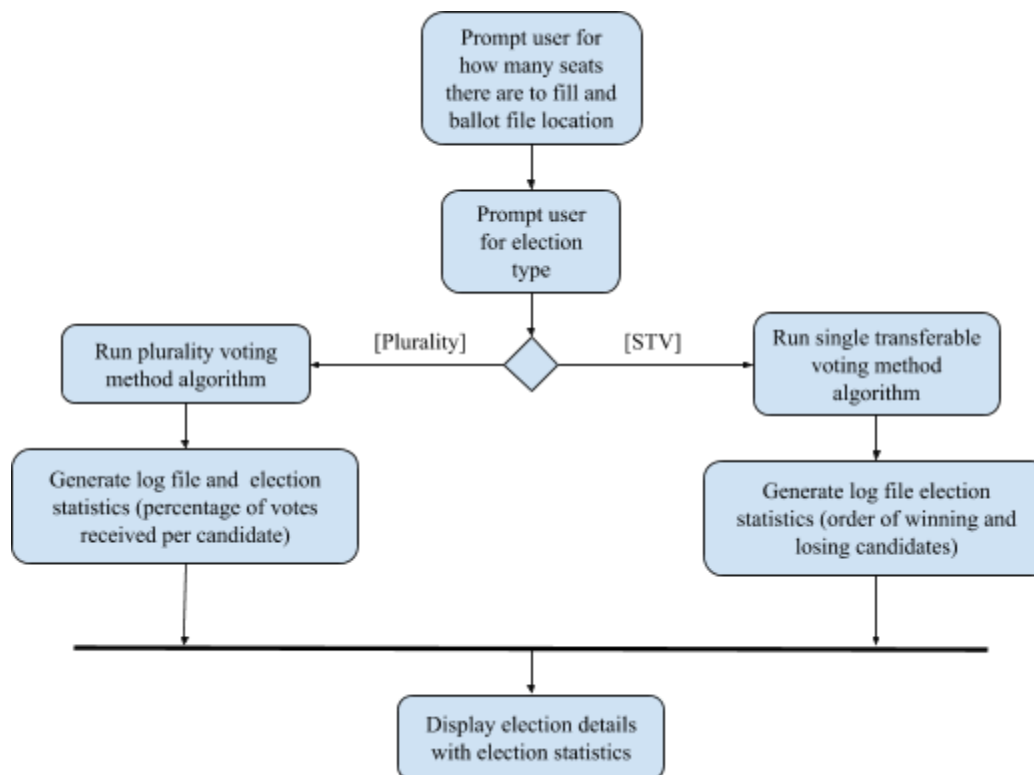
2. Overall Description

2.1 Product Perspective

This software was developed with the intention of being a system which counts and organizes votes to determine the winner(s) of a small scale election. It eliminates the need for anyone to manually count and organize completed ballots. This software was developed as a new self-contained product.

2.2 Product Functions

This software will allow election officials to process the ballots cast in an election in order to determine the winning candidate(s). It is capable of performing both the plurality voting method and the single transferable voting (STV) method using the Droop quota. Once the winners are calculated, the details of the election, including winners/losers and election statistics, are displayed to the user.



2.3 User Classes and Characteristics

The User class that we anticipate using this voting system would primarily be election officials and the people responsible for monitoring and ensuring valid voting results are captured. The election officials are the most important user class for this system because they are responsible for capturing

voting results and this system will provide those results. The system must be able to be multifaceted and be capable of satisfying requirements directly pertaining to the user class such as being able to perform STV method or the plurality voting method depending on what is required by the election officials. Developers/testers may also use this software but will not have any special permissions or a different class.

2.4 Operating Environment

The program will run on CSE Labs Machine with a Java compiler.

2.5 Design and Implementation Constraints

This software will be coded in Java and the user interface will be written in English. The software and relevant hardware must be able to process up to 100,000 ballots in under 5 minutes.

2.6 User Documentation

The user documentation for this software will be a help menu that describes how to use the software. This can be accessed by clicking the “Help” button.

2.7 Assumptions and Dependencies

It is assumed that the users have been correctly trained on how to use the software. The software will also assume that there are no mistakes in the input files and the files are Windows CSV files. We assume that users will provide valid input for the number of seats, voting algorithm type, and location of files. We also assume that the security of the voting is ensured at the voting precinct itself. Additionally, the software assumes that the user will only need to turn ballot shuffling off for the STV algorithm for testing purposes, so it is only permitted as a command line argument.

3. External Interface Requirements

3.1 User Interfaces

The user interface of this software will be textual. The software will prompt the user on what to input and when, as well as how to run the software. The help menu can be opened at any time by typing “Help.”

When running the voting algorithm the user will be prompted to enter the following information: the names of the files containing the ballots, how many seats there are to fill, which voting algorithm is to be used, plurality or STV, if STV is selected the user will be asked if the ballots should be shuffled.

After the election has finished running, the results will also be displayed on the screen in a textual format. This will include the location of the audit file.

3.2 Hardware Interfaces

There are no required hardware interfaces related to this software.

3.3 Software Interfaces

This software requires Java to be installed on the system.

3.4 Communications Interfaces

There are no required communication interfaces related to this software.

4. System Features

4.1 Input Seats to Fill

Name	Input Seats to Fill
ID	UC1
Description	The user inputs the number of seats to fill in the election.
Actors	Election Official; Tester
Organizational Benefits	By inputting the number of seats to fill into the voting software the candidates can be chosen and robustly assigned to each seat based on their rankings with assurance and accuracy.
Frequency of Use	Once per election
Triggers	User starts software
Preconditions	<ul style="list-style-type: none">• Software is open
Postconditions	<ul style="list-style-type: none">• The number of seats in the election that need to be filled has been entered.
Main Course	<ol style="list-style-type: none">1. User is prompted to enter the number of seats2. User types the number of seats3. User hits “Enter” on the keyboard

Alternate Courses	Not applicable.
Exceptions	User types an invalid input See MC 1

4.2 Input Voting Algorithm

Name	Input Voting Algorithm
ID	UC2
Description	The user has a choice between running a plurality style election or an STV style election. The user must enter this into the software.
Actors	Election Official; Tester
Organizational Benefits	This allows the software to be multifaceted and work for a variety of voting situations where either voting method could be used to determine who is elected.
Frequency of Use	Once per election
Triggers	The user hits the “Enter” on the keyboard for the number of seats prompt
Preconditions	<ul style="list-style-type: none"> • The software is running • The number of seats has been entered
Postconditions	<ul style="list-style-type: none"> • The voting algorithm that the user wants to use has been selected.
Main Course	<ol style="list-style-type: none"> 1. User is prompted to enter the type of election 2. User types either “Plurality” or “STV” 3. User hits “Enter” on the keyboard
Alternate Courses	Not applicable.
Exceptions	User types something other than “Plurality” or “STV” See MC 1

4.3 Input File Location

Name	Input File Location
ID	UC3
Description	User enters the name of the folder where the ballot files are all located.
Actors	Election Official; Tester
Organizational Benefits	This is an essential part of running the software correctly and ensuring that proper voting data can be found and pulled into the voting software from a viable location to obtain accurate results.
Frequency of Use	Once per election
Triggers	The user hits the “Enter” on the keyboard for the type of election prompt
Preconditions	<ul style="list-style-type: none">• The software is running• The number of seats has been entered• The type of election has been entered
Postconditions	<ul style="list-style-type: none">• The name of the folder containing the ballot files has been entered.
Main Course	<ol style="list-style-type: none">1. User is prompted to enter the location of the file(s)2. User types valid location3. User hits “Enter” on the keyboard
Alternate Courses	Not applicable.
Exceptions	User types an invalid location See MC 1

4.4 Help Menu

Name	Help Menu
ID	UC4
Description	The user is able to select a help menu that will open a window containing information on how to run the program.

Actors	Election Official; Tester
Organizational Benefits	Reduces wait time for needing help and increases ease of use and efficiency
Frequency of Use	As many times as necessary
Triggers	User selects “Help” button
Preconditions	<ul style="list-style-type: none"> • The software is running
Postconditions	<ul style="list-style-type: none"> • The help information, which tells the user how to run the program, has been printed to the screen.
Main Course	<ol style="list-style-type: none"> 1. User selects the help menu option. 2. Instructions to run the election are displayed to the user. 3. User hits the “Enter” key to return to the main menu once they have finished reading the instructions.
Alternate Courses	Not applicable.
Exceptions	

4.5 Run Election

Name	Run Election
ID	UC5
Description	User triggers the voting algorithm to count votes and determine the winners.
Actors	Election Official; Tester
Organizational Benefits	It is integral to the importance of determining a winning candidate or candidates and is the essence of the software’s mission to provide a quick and seamless way to calculate who would win based off of two distinct voting methods.
Frequency of Use	Once per election
Triggers	User hits ”Enter” on the keyboard
Preconditions	<ul style="list-style-type: none"> • All required input has been entered (number of seats, type of election, and file location)

Postconditions	<ul style="list-style-type: none"> The election has been completed and the number of selected seats have been filled.
Main Course	<ol style="list-style-type: none"> User hits the run button. Election algorithm is run. Results are displayed (see UC6)
Alternate Courses	Not applicable.
Exceptions	<p>Election algorithm fails during the election. User is notified that the software has failed to declare winners. User is instructed to restart the software and try again. Program terminates.</p>

4.6 View Election Results

Name	View Election Results
ID	UC6
Description	After all the votes have been counted and winners have been declared, the results of the election are displayed
Actors	Election Official; Tester
Organizational Benefits	Allows the user to see a comprehensive summary of the election that they just ran.
Frequency of Use	Once per election
Triggers	The election software finishes election successfully
Preconditions	<ul style="list-style-type: none"> The election software has finished running
Postconditions	<ul style="list-style-type: none"> The election results are displayed on the screen <ul style="list-style-type: none"> The type of election (STV/plurality), the number of ballots, the number of seats, and the number of candidates. List of candidates in order, with their respective percentages, which will separate losers and winners
Main Course	<ol style="list-style-type: none"> Election results are displayed
Alternate Courses	Not applicable.

Exceptions	Showing the election results fails due User Interface
------------	---

4.7 Create Log File

Name	Create Log File
ID	UC7
Description	An auditable log file is generated as the software is run that contains step by step documentation of the algorithm as it was run.
Actors	Election Official; Tester
Organizational Benefits	The user has an automatically generated report that acts as an audit for the election.
Frequency of Use	Once per election
Triggers	Voting algorithm has started
Preconditions	<ul style="list-style-type: none"> The election software has been started
Postconditions	<ul style="list-style-type: none"> An auditable log file has been generated in the same location as the program, containing information regarding the election that was just run.
Main Course	<ol style="list-style-type: none"> Voting algorithm begins. Log file is created in the folder that ballot files are located. As the voting algorithm runs, the candidate each ballot goes to is recorded and winners/losers are recorded as they are declared.
Alternate Courses	<p>If STV voting type is chosen:</p> <ol style="list-style-type: none"> Voting algorithm begins. Log file is created in the folder that ballot files are located. Shuffle method of the ballots is recorded. See Main Course step 3.
Exceptions	<p>Election algorithm fails during the election.</p> <p>Failure of the election algorithm is recorded at the point of failure in log file.</p> <p>System error message is recorded in log file.</p>

4.8 Turn Shuffle Option Off

Name	Turn Shuffle Option Off
ID	UC8
Description	Users are able to turn off shuffling of votes in a single transferable voting method election in order to test if software runs correctly.
Actors	Tester
Organizational Benefits	Allows the developers to test if their algorithm accurately counts votes to ensure software quality.
Frequency of Use	As often as it takes to validate software runs correctly.
Triggers	“--shuffle-off” flag is included in input for the election type Ex: STV --shuffle-off
Preconditions	<ul style="list-style-type: none"> • The software is running • The number of seats has been entered
Postconditions	<ul style="list-style-type: none"> • The portion of the software that shuffles the ballots has been turned off.
Main Course	<ol style="list-style-type: none"> 1. User enters “--shuffle-off” flag after the input for the election type on the command line. 2. User hits “Enter” 3. See UC3
Alternate Courses	User wants shuffle on <ol style="list-style-type: none"> 1. See UC2
Exceptions	User types incorrect additional input <ol style="list-style-type: none"> 1. See UC2 Main Course Step 1

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software will be able to process an election in under 5 minutes. The election will receive no more than 100,000 ballots.

5.2 Safety Requirements

There are no safety requirements for this software.

5.3 Security Requirements

There are no security requirements for this software.

5.4 Software Quality Attributes

- Availability - ensure the software can be used by any election official wanting to determine a winner of an election.
- Correctness and reliability- the software must correctly determine the winner/s of an election with no errors.
- Portability- must be able to be downloaded as an application onto any desktop and run using a csv file holding voting information.
- Reusability - the software must be constantly up-to-date and have the ability to be used multiple times for various elections.
- Testability - The software must be able to be tested by a user by running a test file and turning the shuffle option off.
- Usability - The software must be easy to use so that an election official with no prior technical experience can be trained in 15 minutes on how to use the software.

5.5 Business Rules

The election officials will be the only user class to use the voting system thus they will be able to perform all functions of the voting software when an election is taking place. Testers will also use the software however will have the same permissions as election officials.

6. Other Requirements

Appendix A: Glossary

- **STV** (Acronym): Single Transferable Voting; a ranked choice system of voting that utilizes Droop quota to declare candidates winners.
- **Droop Quota**: The smallest number that guarantees that no more candidates can reach the quota than the number of seats available to be filled, calculated by the following equation:

$$\left\lceil \left(\frac{\text{numberOfBallots}}{\text{numberOfSeats}} + 1 \right) \right\rceil + 1$$

Appendix B: Analysis Models

Not applicable.

Appendix C: To Be Determined List

Not currently applicable.