Software Requirements Specification

for

Voting System

Version 1.0

Prepared by Team 18
Linh Duong (duong172)
Thomas Haupert (haupe033)
Henry Huynh (huynh407)
Roshina Mohamed Rafee (moha0797)

University of Minnesota

February 2021

Table of Contents

Introduction 1
- **Purpose** 1
- **Document Conventions** 1
- **Intended Audience and Reading Suggestions** 1
- **Product Scope** 1
- **References** 2

Overall Description 2
- **Product Perspective** 2
- **Product Functions** 2
- **User Classes and Characteristics** 2
- **Operating Environment** 3
- **Design and Implementation Constraints** 4
- **User Documentation** 4
- **Assumptions and Dependencies** 4

External Interface Requirements 5
- **User Interfaces** 5
- **Hardware Interfaces** 6
- **Software Interfaces** 6
- **Communications Interfaces** 7

System Features 7
- **Instant Runoff Voting Support** 7
- **Open Party List Voting Support** 7
- **Displaying of Results** 7
- **Audit File Production** 8
- **Media Report File Production** 8
- **Tie Vote Handling** 8

Other Nonfunctional Requirements 9
- **Performance Requirements** 9
- **Safety Requirements** 9
- **Security Requirements** 9
- **Software Quality Attributes** 9
- **Business Rules** 10

Other Requirements 10

Appendix A: Glossary **10**

Appendix B: Analysis Models **10**

Appendix C: To Be Determined List **11**

Appendix D: Use Cases **11**

Determine File Name **11**

Read In File **12**

Process File **13**

Run Instant Runoff **14**

Run Open Party List **15**

Independent Candidates **16**

Coin Flip on Tie **17**

Print Results to Screen **18**

Produce Audit File **19**

Produce Media Report File **20**

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to provide a comprehensive description of the Voting System software. Further on, we will explain the functionality of the system, the interfaces present, the system's response to limitations, constraints under which the system will act, and the feedback mechanisms. The voting system will be using Instant Runoff Voting and Party List voting using Open Party List. This document is our first release and is intended for project purposes in CSCI 5801. This document describes only part of the system; in particular, it describes the tallying of votes and the final results of an election. The actual casting of the votes is outside the scope of this document.

1.2 Document Conventions

This document was prepared using the Institute of Electrical and Electronics Engineers (IEEE) template for Software Requirements Specification (SRS) documents.

Prompts provided by the program will be displayed in `10-point Roboto Mono.`
Inputs to the program that are typed in by the user will be displayed in ***`Italicized and bolded 10-point Roboto Mono.`***
Keyboard buttons that the user can press are surrounded with a less-than (<) and greater-than (>) symbol, such as in <Enter> or <A>.

1.3 Intended Audience and Reading Suggestions

This document is intended for the purpose of informing both potential users and or proponents of the voting program, as well as those who intend to test, document, or implement the program.

For a general overview of the software, Section 1.4: Product Scope, Section 2: Overall Description, and Section 4: System Features are the most relevant chapters.

1.4 Product Scope

This software system will be a voting system that is capable of performing two types of voting: Instant Runoff Voting (plurality/majority type) and party list voting using Open Party List Voting (proportional voting). The purpose of this system is to count the number of votes based on chosen voting algorithms, produce an audit file with election information that replicates the election progress itself, and display the name(s) of the winner(s) as well as other pertinent information both onscreen and in a media report file.

1.5 References

Douglas, J. A. (n.d.). Instant Runoff Voting. FairVote. Retrieved from https://www.fairvote.org/instant_runoff_voting_no_substitute_for_pr

FairVote. (n.d.). How Proportional Representation Elections Work. Retrieved from https://www.fairvote.org/how_proportional_representation_elections_work

Oracle. (n. d.). How to Write Doc Comments for the Javadoc Tool. https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html

2. Overall Description

2.1 Product Perspective

This software is intended as a replacement for current voting system software. Its primary purpose is to determine the winner of an election in two alternative voting systems: Instant Runoff Voting and Open Party List Voting. It supports voting results in the form of a CSV file as input to the system. An audit file and media report file in the form of a TXT document are output from the system.

This software is not intended to be used in any part of the actual voting process, such as in the casting of votes or the processing of individual ballots. It is to instead be used after an election has been conducted, where voting results will be input to the software to obtain the winner(s) of the election.

2.2 Product Functions

The product will allow an election official to calculate the result of an election conducted as an Instant Runoff Vote or Open Party List Vote. It will allow for the insertion of an election file in the form of a CSV that contains the ballots of the election. The product will also allow for the creation of two output files: an audit file for the election officials and testers, as well as a media report file. An audit file is a formal record documenting how every single ballot gets distributed (in case of Instant Runoff Vote), replicating the election itself, and producing the results of the election. This audit file will be used by the testers to check the result and keep records. The media report file will show the outcome of the election and is intended for dissemination to the media.

2.3 User Classes and Characteristics

Election officials: This user class includes those who are responsible for the use of the software to tally up the votes of an election. They are the most important user class as they are the end-user of the product. They have a high privilege level, so they are the ones responsible for the actual running of the software in a real-world election scenario. They have a low technical expertise, so the software should be written so that it is easily accessible. They will use the software multiple times a year when an election or special election is held.

Media: This user class does not use the software, but instead receives a media report file that is output by the software. This user class has a low technical expertise, so the media report file should be organized in a way that is easy to understand.

Programmers: This user class includes the developers of the software. They have high technical expertise and will thus be responsible for the maintenance of the software and the fixing of any bugs or errors that are found. They will also be responsible for adding additional features or functionality to the software as needed.

Testers: This user class includes those who are responsible for ensuring that the software behaves as expected. They have medium technical expertise and will send any bug reports to the programmer for fixing.

2.4 Operating Environment

This software will be used independently for counting votes after an election has been conducted, so it does not need to work with any other software components or applications aside from Java. It is confirmed to work with these minimum system requirements:

OS	Windows 10
MACHINE	Dell Precision 3630
PROCESSOR	Intel Core i7-4790 @ 3.6 GHz (x4)
MEMORY	32 GB RAM
HARD DRIVE	1GB free space
JAVA VERSION	OpenJDK 11.0.9.1

OS	Ubuntu 20.04
MACHINE	Dell XPS 8910
PROCESSOR	Intel Core i7-6700 @ 3.4 GHz (x4)
MEMORY	16 GB RAM
HARD DRIVE	1GB free space
JAVA VERSION	OpenJDK 11.0.9.1

OS	Mac OS X
PROCESSOR	Intel Core i5 @ 3.2 GHz (x4)
MEMORY	16 GB RAM
HARD DRIVE	1GB free space
JAVA VERSION	OpenJDK 11.0.9.1

2.5 Design and Implementation Constraints

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

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

Technology constraints: The software requires Java (version 11) to be installed on the computer in order for it to run.

Language constraints: The software will be written in the Java programming language. The software prompts and documentation will be written in the English language.

Databases to be used: The database to be used is an audit file (CSV formatted file that is exported from Excel). The audit file is a strict requirement and the first line of the file will provide the type of voting.

2.6 User Documentation

Documentation for the end-user will be provided along with the program in the form of a TXT document with basic instructions on how to operate the software. The actual software itself will also provide instructions in a step-by-step manner. The software will also have documentation intended for developers and maintainers. It will be provided in the form of a Javadoc. The Javadoc will be written according to the standards provided by Oracle. It will be available inside of the code and as a separate HTML document.

2.7 Assumptions and Dependencies

The project depends on certain assumptions about how the election was conducted. The election must not allow write-in candidates, as the software does not support it. The election must also use either of the two voting systems: Instant Runoff Voting or Open Party List Voting. The ballots must all have been preprocessed ahead of time to ensure that voters have filled them out correctly and placed in the election file. The election file must also be located in the same directory as the software.

If the election is run using *Instant Runoff Voting*, the election file must follow these guidelines:

- ❏ The first line of the file must contain “IR” to indicate the type of election.
- ❏ The second line must be an integer greater than 0 that represents the number of candidates.
- ❏ The third line must contain the names of the candidates, followed by the name of their party in parentheses, with commas separating the candidates.
- ❏ The fourth line must be an integer greater than 0 that represents the number of ballots in the file.
- ❏ Each following line is a ballot that contains the ranking of each candidate, counting up from 1, and separated with commas.

If the election is run using *Open Party List Voting*, the election file must follow these guidelines:

- ❏ The first line of the file must be “OPL” to indicate the type of election.
- ❏ The second line must be an integer greater than 0 that represents the number of candidates.
- ❏ The third line must contain - in brackets - both the name and party of the candidate separated by a comma. Additional candidates can be added to the line separated by commas.
- ❏ The fourth line must be an integer greater than 0 that represents the number of seats.
- ❏ The fifth line must be an integer greater than 0 that represents the number of ballots.
- ❏ Each following line is a ballot that contains only a single 1, with commas separating the candidates.

3. External Interface Requirements

3.1 User Interfaces

The software will use a text-based interface to obtain the text file from the user. No graphical user interface is available.

Upon starting the program, the software will display this to the terminal:

```
Voting System
Please enter election file name:
```

The software will now wait for the user to input the name of the election file. For example, if the user wishes to load the election file named “test.csv”, they can type:

test.csv

The user can then press <Enter> to tell the program to load the file.
If the file is not found, this error message will be displayed:

```
Error: Election file was not found.
```

The software will then ask the user again to insert the election file name as above.
If the file is found, the software will process the file and then display the results in this form:

```
Winner(s) of election:
Type of election:
Number of ballots cast:
Number of seats:
```

The fields above will be populated with the information from the ballot. “Winner(s) of election” will display the names of each candidate who won the election. “Type of election” will display either “Instant Runoff” or “Open Party List” based on the type of election. “Number of ballots cast” will display an integer representing the total number of votes. “Number of seats” will display an integer representing the number of open seats. Following these results will be a list of each candidate and the amount of votes they got.

After this, the audit file and media report file will be created in the working directory of the program. The audit file will have a unique name based on the current time and date in the format “`audit_MM-DD-YYYY_HH.MM.SS.txt`”. For example, this will show when the program is run on February 1, 2021 at 15:20:50 (or 3:20:50 PM):

```
An audit file has been created: audit_02-01-2021_15.20.50.txt
A media report file has been created: report.txt
```

At this point, the program will exit.

3.2 Hardware Interfaces

The software uses Java to interface between the product and the hardware of the computer. The computer and operating system must be compatible with Java (version 11) for the program to work. Additional system requirements are listed in Section 2.4: Operating Environment.

This software requires the following hardware components to run successfully:

- Keyboard (or other typing peripheral)
- Display device (such as a monitor or television)
- Storage device (such as a hard drive or flash drive)

3.3 Software Interfaces

This software requires Java (version 11). An election file will be input to the program by the user, which will contain information regarding the election such as which candidate each person voted for and what kind of voting format took place. The election file that is entered by the user will be shared across the software to process and obtain information that will be key to calculate the winners of the election. Once the votes have been counted for the specific voting system, a media report file will be produced that will display the winner of the election, the kind of voting system, and the number of votes the candidate secured. In addition to this, an audit file will be produced which will show the calculations that the software performed.

3.4 Communications Interfaces

This software runs entirely locally. No communication interfaces are required.

4. System Features

This section provides a high-level overview of the features of the software. Full use cases are available in Appendix D: Use Cases.

4.1 Instant Runoff Voting Support

4.1.1 Description and Priority

The software is able to support the counting of votes in elections that use Instant Runoff Voting. This feature is of high priority.

4.1.2 Stimulus/Response Sequences

This feature will be activated when the election file has “IR” listed on the heading.

4.1.3 Functional Requirements

IRV-1: The software must be able to read “IR” from the first line of the election file.
IRV-2: If the software instead reads “OPL” on the first line, it will instead use Open Party List Voting.
IRV-3: The software assumes that the first line of the election file will read “IR” or “OPL”.

4.2 Open Party List Voting Support

4.2.1 Description and Priority

This software supports the Party List Voting that uses Open Party List. This feature is of high priority.

4.2.2 Stimulus/Response Sequences

This feature will be activated when the election file has “OPL” listed on the heading.

4.2.3 Functional Requirements

OPL-1: The software must be able to read “OPL” from the first line of the election file.
OPL-2: If the software instead reads “IR” on the first line, it will instead use Instant Runoff Voting.
OPL-3: The software assumes that the first line of the election file will read “IR” or “OPL”.

4.3 Displaying of Results

4.3.1 Description and Priority

The software must be able to display the result of the election to declare the winner(s). This feature is of high priority.

4.3.2 Stimulus/Response Sequences

The winner(s) of the election will be displayed onto a screen/monitor when the system has finished processing the input. The display will follow the structure mentioned in Section 3.1: User Interfaces.

4.3.3 Functional Requirements

DR-1: The software must have finished processing the input file
DR-2: The display must show the name of the winner(s), the number of ballot cast, and the number of votes for every candidate.

4.4 Audit File Production

4.4.1 Description and Priority

The software must be able to produce an audit file for the election officials to verify that the election was counted fairly and accurately. This feature is of high priority.

4.4.2 Stimulus/Response Sequences

The audit file will be created automatically when the election file is input and processed. A prompt on the screen will display when the audit file is created along with the name of the file.

4.4.3 Functional Requirements

AFP-1: The software must be able to successfully read the file.
AFP-2: The software must simulate the results of the redistribution of votes as the processing takes place.

4.5 Media Report File Production

4.5.1 Description and Priority

This software must be able to produce a report to be provided to the media. This feature is of high priority.

4.5.2 Stimulus/Response Sequences

A report of the election will be generated when the system has finished producing an audit file.

4.5.3 Functional Requirements

MRD-1: The software must have finished generating the audit file and displaying the result onto the monitor.
MRD-2: The software must show the name of the winner(s), the number of ballots cast, and the number of votes for every candidate/party.

4.6 Tie Vote Handling

4.6.1 Description and Priority

This software must be able to fairly toss a coin to handle the situation in which the vote is tied. This feature is of high priority.

4.6.2 Stimulus/Response Sequences

The system will toss a coin fairly to choose a winning candidate randomly when there are tie votes between candidates. The winning candidate(s) will be decided and stored for further processing.

4.6.3 Functional Requirements

TVH-1: The software must have successfully begun reading the ballots using Instant Runoff Voting or Open Party List Voting.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software should be able to process 100,000 ballots in under 8 minutes. This is a performance requirement because during election times there will be large amounts of voter turnouts which leads to having enormous amounts of ballots. In order to get the results out in a timely manner it is important to be able to run the program efficiently and quickly.

5.2 Safety Requirements

There are no special safety requirements for this project.

5.3 Security Requirements

There are no security requirements for this project.

5.4 Software Quality Attributes

Correctness: The software should be able to provide the correct result for an election every time. Election files should be tested with the software and should be compared to the result calculated by hand to ensure correctness.

Robustness: The software should be written to ensure that it can run correctly and perform well even with a very high volume of ballots, as it must be able to process 100,000 ballots in under 8 minutes.

Extensibility: The software should be easily changed without affecting other parts of the software. This is because additional functionality may be added to the software in the future, such as support for write-in candidates.

Portability: The software should be designed so that it is able to be easily moved to other environments. The software may in the long-term be implemented as part of an online voting system.

5.5 Business Rules

The election official should be responsible for the running of the software. When the program is finished, an audit file and media report file will be generated. The election officials should review the audit file to ensure that the calculation of the election winner(s) was/were done correctly. The media report file should be sent to members of the press.

6. Other Requirements

There are no additional requirements to add to this section.

Appendix A: Glossary

Term	Definition
Audit file	A TXT file output by the program that contains the calculations that were executed to produce the winner. This includes showing all of the ballots that were added up as well as the results of each redistribution of the votes. This file is intended for use by election officials to ensure the final results are correct.
Election file	The CSV file input to the program that contains the type of election conducted, the number of candidates, the names of each candidate and their party, the number of seats, the number of ballots, and the result of each ballot.
Instant runoff voting (IRV)	A type of voting system in which voters can rank the candidates to ensure their vote is transferred in case their primary choices cannot win the election.
Media report file	A TXT file output by the program that contains the results of the election, including the type of election, seats, candidates, percentages of the vote, and the winner(s) of the election. This file is intended for distribution to the media.
Open party list ballot (OPL)	A type of voting system in which voters vote for a particular party and particular candidates and are elected proportionally.
Write-in	When a voter “writes-in” the name of a person they want elected instead of voting for a candidate on the ballot.

Appendix B: Analysis Models

Analysis models (including UML diagrams) will be added at a later date.

Appendix C: To Be Determined List

There are no to-be-determined references.

Appendix D: Use Cases

Determine File Name

Name	Determine File Name
ID	PRC_001
Description	Determine the file name of the election file.
Actor(s)	Election Official, System
Organizational Benefits	To gather all the data required to announce the winner of the election.
Frequency of Use	Multiple times per year at normal election times and special elections.
Trigger	When the election officials receive an election file after an election has been conducted and the software is run.
Precondition	The election file should be present in the working directory of the software.
Postcondition	The election filename will be stored for later use.
Main Course	1. The system will prompt the user for the file name. 2. User will enter in the file name (see EX1). 3. The system will save the file name that was entered.
Alternate Courses	No alternate courses
Exceptions	EX1 File name entered is not found 1. System prompts user for file name another time. 2. If the file is found, proceed to Main Course 3. 3. If the file is not found, return to EX1.

Read In File

Name	Read In File
ID	PRC_002
Description	Read in the election file.
Actor(s)	Election Official, System
Organizational Benefits	To gather all the data required to announce the winner of the election.
Frequency of Use	The file will be read every time the user enters the name of the file.
Trigger	The software will automatically run this after the “Determine File Name” use case.
Precondition	The file name must have been determined and successfully found.
Postcondition	The election file will be loaded by the program for reading.
Main Course	1. The software will load the file that was named earlier in read-only mode.
Alternate Courses	No alternate courses
Exceptions	No exceptions

Process File

Name	Process File
ID	PRC_003
Description	Process all the information listed in the file and store the information in the software.
Actor(s)	Election Official, System
Organizational Benefits	To process data within the file to help us determine the information that is needed to finalize the results
Frequency of Use	The file will be processed every time the user enters the name of the file and the file is read.
Trigger	The software will automatically run this after the “Read In File” use case.
Precondition	An election file must be loaded by the file for reading. The first line of the election file must read “IR” or “OPL”.
Postcondition	The election file will be loaded by the program for processing information.
Main Course	1. The software will read the first line of the election file to determine the type of election (see AC1, AC2).
Alternate Courses	AC1 If the first line of the election file is “IR”: 1. The software will proceed to run the “Run Instant Runoff” use case. AC2 If the first line of the election file is “OPL”: 1. The software will proceed to run the “Run Open Party List” use case.
Exceptions	No exceptions

Run Instant Runoff

Name	Run Instant Runoff
ID	ELEC_001
Description	For elections of the Instant Runoff type, this will process ballots received in the processed file according to the rules of an Instant Runoff type election.
Actor(s)	Election Official, System
Organizational Benefits	Automation of the Instant Runoff election process.
Frequency of Use	Every time there is an election of the type Instant Runoff.
Trigger	The election results file received is of the Instant Runoff type.
Precondition	The elections results file has been identified and processed, with the election type specified as Instant Runoff.
Postcondition	Two separate files are generated. One audit file which details the process of the automated Instant Runoff election, and one media report file which details the winner and final results of the election.
Main Course	1. The system checks to see if a candidate has received greater than half of the votes. 2. If it does, declares that candidate as the winner and generates the audit and media report files. If a candidate does not have the majority, the program continues to step 3. 3. The current state of the election is stored for the audit file. 4. The candidate with the fewest votes is eliminated and all of the votes of the eliminated candidate are given to the candidate whose name shows up next on their ranked ballot. 5. The system repeats steps 1-4 until a winner is declared. 6. Audit and media report files are produced. 7. The results are printed to screen.
Alternate Courses	AC1 In the case that there is a complete tie at some point in the election where there is no candidate with the fewest votes: 1. The system proceeds to the coin flip process.
Exceptions	EX1 In step 3, if a ballot does not have a next choice: 1. The ballot is tossed and the vote is not given to any candidate.

Run Open Party List

Name	Run Open Party List
ID	ELEC_002
Description	For elections of the Open Party List type, this will process ballots received in the processed file according to the rules of an Open Party List election.
Actor(s)	Election Official, System
Organizational Benefits	Automation of Open Party List election process.
Frequency of Use	Every time there is an election of an Open Party List type.
Trigger	The election results file received is of the Open Party List type.
Precondition	The elections results file has been identified and processed, with the election type specified as Open Party List.
Postcondition	Two separate files are generated. One audit file which details the process of the automated Open Party List election, and one media report file which details the winner and final results of the election.
Main Course	1. The system determines the quota by dividing the total number of votes by the number of seats available. 2. The system then tallies the total votes received by each party. 3. The total votes of each party are divided by the quota and a seat is given to that party for each whole number produced. If not all seats are given, see AC1. 4. For each seat given to a party, the candidates with the highest votes are chosen until all seats are filled. 5. Audit and media report files are produced. 6. The results are printed to screen.
Alternate Courses	AC1 In the case that not all seats are given after the initial counts: 1. The seats are awarded to the parties closest to the next whole number (highest remaining votes). If there is a tie for the majority, proceed to the coin flip process. 2. The system proceeds to Main Course 4.
Exceptions	No exceptions

Independent Candidates

Name	Independent Candidates
ID	ELEC_003
Description	For the purpose of an Open Party List election, the independents are grouped into one party.
Actor(s)	Election Official
Organizational Benefits	Due to the nature of how seats are given based on the number of total votes given to a party, the independents must be grouped together in order for them to obtain seats.
Frequency of Use	This will occur whenever there is an Open Party List election with independent candidates.
Trigger	There is an Open Party List election with independents running.
Precondition	All of the votes have been counted and party listings created.
Postcondition	The independents are grouped together to form the “Independents” party for the purpose of the Open Party List election process.
Main Course	1. All of the independent candidates are grouped into one party called the “Independent party”.
Alternate Courses	No alternate courses
Exceptions	EX1 In the case that there is only one independent candidate: 1. The independent party is still created, with that candidate as the sole member.

Coin Flip on Tie

Name	Coin Flip on Tie
ID	ELEC_004
Description	Toss the coin fairly to randomly select a winner if there is a tie in the votes.
Actor(s)	Election Official, System
Organizational Benefits	To keep the counting process fair without repeating the election again.
Frequency of Use	The coin flip is used every time there is a tie in regards to the election process.
Trigger	There is a tie vote cast between the candidates or between parties.
Precondition	The input file has been processed; the candidates have been assigned with the number of votes cast.
Postcondition	The name of the chosen winner will be stored for further processing.
Main Course	1. The system checks the candidates whose votes are tied then group these candidates together. 2. The system checks the number of candidates (winners) needed to be chosen from this group. If there is more than 1 candidate needed to be chosen, proceed to AC1. 3. The system then generates a random seed to choose one candidate out of this group. 4. The candidate chosen by the random seed is stored and returned as a winner within the group.
Alternate Courses	AC1 In the case that the system is required to choose more than 1 candidate in the group of candidates with tie votes: 1. The system generates a random seed and chooses one candidate out of this group. The chosen candidate is stored in another group for winners. 2. The system then removes the chosen candidate from the tie-vote group. 3. The system repeats steps 1 and 2 until the system satisfies the number of candidates needed to be chosen. 4. The system returns the winner group with the stored name of candidate(s).
Exceptions	No exceptions

Print Results to Screen

Name	Print Results to Screen
ID	OUT_001
Description	Prints the results of an election to standard output.
Actor(s)	System
Organizational Benefits	This process is useful when the results of an election need to be visible. This is also useful in the debugging process and to ensure there are no errors.
Frequency of Use	Everytime an Instant Runoff or Open Party List election process is run, the final results will be printed to screen.
Trigger	The final step in the Instant Runoff and Open Party List after producing media and audit files are to print the results to screen.
Precondition	The Instant Runoff or Open Party List process has finished.
Postcondition	The results of the election are printed out to standard output.
Main Course	1. The system determines the type of election. 2. If it was an Instant Runoff election, displays a list of candidates along with the votes at each count. 3. Otherwise, in the case of Open Party List elections, displays a list of the parties along with total votes, seats allocated initially, final seat allocation, and selected candidates from each party.
Alternate Courses	No alternate courses
Exceptions	EX1 If the election process was run but an error occurred: 1. System prints out “An error occurred in the election process.”

Produce Audit File

Name	Produce Audit File
ID	OUT_002
Description	After an election process concludes, an audit file detailing the steps taken by the system in generating the results is produced.
Actor(s)	System
Organizational Benefits	The audit file is a highly important file which allows the election process to be visible and thus validated. Should the results of an election be challenged, the audit file is a key piece of evidence.
Frequency of Use	At the end of every election process run by the system, an audit file will be produced.
Trigger	The produce audit file process is triggered at the end of every election process.
Precondition	An election process of either type concludes.
Postcondition	An audit file is produced which details the steps taken by the system in determining the winner of the election.
Main Course	1. The system determines which election type the audit file will be for. 2. During the election process, the initial, intermediary, and final states of the process were stored. The system compiles these states into a readable file that displays how the election process occurred according to the format of the election type.
Alternate Courses	No alternate courses
Exceptions	EX1 If the election process was run but an error occurred: 1. System prints out “An error occurred in the election process.”

Produce Media Report File

Name	Produce Media Report File
ID	OUT_003
Description	After an election process concludes, a media report file will be produced which details the final winner and results of the election.
Actor(s)	System
Organizational Benefits	The media report is a document ment to report the final results of an election process in a readable and concise manner.
Frequency of Use	At the end of every election process run by the system, a media report file will be produced.
Trigger	The produce media report process is triggered at the end of every election process.
Precondition	An election process of either type concludes.
Postcondition	A media report is produced which details the final results of the election.
Main Course	1. The system determines what type of election the media report is being produced for, to determine the format. 2. The election results are shown according to the format.
Alternate Courses	No alternate courses
Exceptions	EX1 If the election process was run but an error occurred: 1. System prints out “An error occurred in the election process.”