Software Requirements Specification

for

Ballot Processor

Version 0.0 approved

Prepared by Bek Allenson, Perrie Gryniewicz, Matthew Johnson, and Logan Watters

CSCI 5801

October 6th 2023

Table of Contents

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 **2**
2.5 Design and Implementation Constraints **2**
2.6 User Documentation **2**
2.7 Assumptions and Dependencies **3**
3. External Interface Requirements **3**
3.1 User Interfaces 3
3.2 Hardware Interfaces **3**
3.3 Software Interfaces **3**
3.4 Communications Interfaces **3**
4. System Features **3**
5. Other Nonfunctional Requirements **3**
5.1 Performance Requirements 3
5.2 Safety Requirements 4
5.3 Security Requirements **4**
5.4 Software Quality Attributes **4**
5.5 Business Rules **4**
6. Other Requirements **4**

Revision History

Name	Date	Reason For Changes	Version
Original Version	10/6/23	No changes made. Document completed.	0.0

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of a standalone, new Ballot Processor with version number 1.0. The system will use instant runoff voting (IR) and open-party list (OPL) voting algorithms to determine a winner. The document will describe the use cases, constraints, and requirements for the system.

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

Succeeding this introduction, the document discusses a high level overview of the system, external interface requirements, system features, and nonfunctional requirements. This document is intended for:

- Typical users of the system, such as election officials, who want to use Ballot Processor to process election results.
- Developers, who are consulting this document for guidance on how to implement Ballot Processor.
- Testers, who are stress testing Ballot Processor to ensure its correctness.

1.4 Product Scope

The software we are creating will process two types of election algorithms: Instant Runoff and Open Party List. The system will count ballots and distribute them according to the algorithm to declare an election winner. A user will input a file name for the system to process ballots. Based on the ballots, candidates will be reordered and removed to eventually produce election results. This will decrease the amount of time taken to process an election and allow for quick results.

1.5 References

- IEEE Template for System Requirement Specification Documents. Retrieved from CSCI5801 Canvas page.
- UseCases_Team5.pdf. Stored in the team Github repository. Holds all use cases.

2. Overall Description

2.1 Product Perspective

Ballot Processor is a new program, developed for use by election officials to process ballots and produce election results in a transparent, timely manner. It can handle ballots for elections in the Instant Runoff or Open Party Listing formats.

2.2 Product Functions

- Input and process ballot CSV file.
- Determine results of an Open Party List (OPL) voting election.
- Determine results of an Instant Runoff (IR) voting election.
- Generate audit file containing OPL or IR election results.
- Display results of the election and election information to the user.

2.3 User Classes and Characteristics

- Typical users, such as election officials, who want to use Ballot Processor for running either an Instant Runoff algorithm or an Open Party List algorithm on a ballot file
- Programmers, responsible for creating and maintaining Ballot Processor
- Testers, responsible for creating and executing test cases to ensure that the program behaves as intended.

2.4 Operating Environment

The environment for this software product will be any computer and will utilize Java. This is version 1 of the software.

- Windows
- Linux
- Mac OS

2.5 Design and Implementation Constraints

Ballot Processor is developed in Java. Ballot Processor must be able to run 100,000 ballots in under 8 minutes. The input file will always be in a CSV format. Program must display the results of the election to the user, and produce an audit file with election results and details.

2.6 User Documentation

Ballot Processor will guide the user to input the name of the .csv ballot file. If the file name is invalid, the program will inform the user of this and request another filename entry. There is nothing else required of the user.

2.7 Assumptions and Dependencies

The system assumes the election will be an Instant Runoff or Open Party List election. The system depends on the election input file being in the correct format and the ballot counting software to be accurate.

3. External Interface Requirements

3.1 User Interfaces

The user interface for Ballot Processor will be fully confined to the command line. Ballot Processor does not have a GUI, and users will interact with the program solely through command line input.

3.2 Hardware Interfaces

There are no interface requirements with hardware. Since the system will be written in Java, all machines with a command line interface can run the system.

3.3 Software Interfaces

Ballot Processor requires a current version of Java to be installed on the system.

3.4 Communications Interfaces

This product should be accessible to all voters, election officials, and testers. The output of the data will be straightforward and produce an audit file including election results and details. The main source of communication is the command line for user input and the audit file produced as a .txt file.

4. System Features

See the functional requirements and use cases in UseCases_Team5.pdf

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Ballot Processor is designed to run on a CSE Labs machine. It will be able to process 100,000 ballots in under 8 minutes.Safety Requirements

5.2 Safety Requirements

None for this system.

5.3 Security Requirements

The requirements of the software must maintain the confidentiality of votes to stay valid. The ballots should not have any disclosed information about the voter or who they have voted for. The file contains the votes for the candidates and can't share the ranking of candidates until the final calculation. The only information that should be shared is within the created audit file which discloses the election results. Ballot Processor does not modify the ballot file in any way, guaranteeing protection from election fraud.

5.4 Software Quality Attributes

The software will produce the correct output of the election based upon the election type selected and will not modify the ballot input file. The produced audit file will be portable between devices and will provide the correct derivation of election results.

5.5 Business Rules

Ballot Processor can be run by any election official, and that individual will have access to the full functionality of the program, including selecting which election file to process, and accessing the election results. They will have to have the ability to view and receive the election file.

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

OPL: Open Party List Voting.
IR: Instant Runoff Voting.

Appendix B: Analysis Models

None.

Appendix C: To Be Determined List

None.