IR Activity: https://lucid.app/lucidchart/invitations/accept/e6250e98-8aca-4b55-a37f-5507ba1681d4
OPL Activity: https://lucid.app/lucidchart/invitations/accept/bf2878ac-cf0a-47f2-8722-c25c6009289e
Class Diagram: https://lucid.app/lucidchart/invitations/accept/994b2b04-04f0-4bb1-9063-3b8ae400f39b
Sequence Diagram:

# Software Design Document (SDD) Template

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and data necessary for the implementation phase. The SDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage, i.e. the detailed design stage, more detailed data structures are defined and algorithms are developed for the defined architecture.

This template is an annotated outline for a software design document adapted from the IEEE Recommended Practice for Software Design Descriptions. The IEEE Recommended Practice for Software Design Descriptions have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. For your own information, please refer to IEEE Std 10161998 [1] for the full IEEE Recommended Practice for Software Design Descriptions.

[1] http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieeeSDD.pdf

(Team Name)

**(Project Title)**

Name (s): Eileen Campbell, Maranda Donaldson, Hazel Dunn, Olivia Hansen
Lab Section:
Workstation:

Date: (mm/dd/yyyy)

**TABLE OF CONTENTS**

**Olivia Eileen Hazel Maranda**

## 1. INTRODUCTION

### 1.1 Purpose

Identify the purpose of this SDD and its intended audience. (e.g. “This software design document describes the architecture and system design of XX. ….”).

This document contains the design specifications for our Voting System. The expected audience is the programmers working on the implementation of the system.

### 1.2 Scope

Provide a description and scope of the software and explain the goals, objectives and benefits of your project. This will provide the basis for the brief description of your product.

This document contains a complete description for the design of a voting system. This system can handle analyzing data for both an Instant Runoff or Open Party List election. It will take in data for a specific election and export election results to a media file, an audit file, and the terminal. The program is written in Java and all user interaction will be through a computer terminal. Using this program drastically reduces human error and the amount of time it takes to

There are no security precautions. Any user with a copy of the program has the authority to obtain election results.

### 1.3 Overview

Provide an overview of this document and its organization.

This document will be organized into 8 sections. The first section will contain an overview of the whole system - its purpose, what the goals of the program are and what it does, references to any outside material that was used, and definitions that may be needed for the reader to understand the context.

The second section will focus on an overview of the whole system and how it will operate. Compared to the previous section, it will get into more detail about the software itself, such as the descriptions of the functionality, context, and design of the program.

Section three will go into more detail about the architecture of the system, how tasks were split up into sections, and why it was designed this way. It will explain how responsibilities of the system were divided into subsystems, and will provide diagrams for explanation.

The fourth section will talk about the data itself, and how it is transformed into data structures. It will describe how the major data are stored, processed and organized.

Section number five will solely focus on the component design, and will go into depth about what each of the components does in the system. Visual aid such as diagrams or examples like pseudocode will be used for explanation.

The sixth section will focus less on the details and operations of the systems themselves, and will instead focus on the user and how the system works with them. It will detail how the system will benefit the user, how it will function for them and their expected features, how they will use it, and how the system will interact with the user through displays and feedback. The section will also provide screenshots of an example of what the program display will look like.

Section seven will be a simple matrix/table that goes through each requirement that the user and designers specified for the program in the SRS document, and traces components and data structures of the system to them.

Section number eight will be appendices that provide any additional information that could aid the user in understanding this document.

### 1.4 Reference Material

*This section is optional.*

List any documents, if any, which were used as sources of information for the test plan.

Instant Runoff Voting Information:
https://www.fairvote.org/glossary

Part List Voting Information:
https://www.fairvote.org/how_proportional_representation_elections_work

Software Designment Document Template (provided by Dr. Shana Watters):
https://drive.google.com/file/d/1b8HCb5cIb10WkQvJl4IVux19gK_NbZjx/view?usp=sharing

Writing the SDD - Assignment Description (provided by Dr. Shana Watters):

### 1.5 Definitions and Acronyms

*This section is optional.*

Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SDD. These definitions should be items used in the SDD that are most likely not known to the audience.

- Instant Runoff Voting (IR): All candidates are listed on the ballot and voters rank candidates in order of their preference.
- Open Party List Voting (OPL): Voters are presented an unordered list of candidates chosen in party primaries. Voters cast a vote for individual candidates. This vote counts for the specific candidate as well as for the party. Number of seats are delegated by which party got the most votes and which candidates were the most popular in each party.
- CSV File: Comma Separated Values. A plain text file that contains a list of data. Every value separated by a comma.

## 2. SYSTEM OVERVIEW

Give a general description of the functionality, context and design of your project. Provide any background information if necessary.

This system's primary purpose is to analyze ballot data to determine an election's results. In order to accomplish this, the system will be able to read in data from a CSV file, store this data in an efficient manner, run one of two specified algorithms to determine the election results, and print the results to multiple files/outputs. To read in data from a given CSV file, the system will prompt the user to input the file name. While getting user input, the system will also prompt the user to see what to call the two output files. After getting all user inputs, the system will then read in all necessary data from the CSV file, using the first line of the file to determine if it will run the IR or OPL voting algorithm. While running the algorithms, the system will continually print to the audit output file to show the progress of the election. Once the algorithms have completed, a summary of results will be printed to the screen and to a media report.

## 3. SYSTEM ARCHITECTURE

### 3.1 Architectural Design

Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high level overview of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Don't go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram if required.

### 3.2 Decomposition Description

Provide a decomposition of the subsystems in the architectural design. Supplement with text as needed. You may choose to give a functional description or an object oriented description. For a functional description, put top level data flow diagram (DFD) and structural decomposition diagrams. For an OO description, put subsystem model, object diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here.

### 3.3 Design Rationale

Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn't choose them.

## 4. DATA DESIGN

### 4.1 Data Description

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized. List any databases or data storage items.

CSV to Objects? -> Objects stored in ArrayLists?

### 4.2 Data Dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

## 5. COMPONENT DESIGN

In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.

## 6. HUMAN INTERFACE DESIGN

### 6.1 Overview of User Interface

Describe the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

### 6.2 Screen Images

Display screenshots showing the interface from the user's perspective. These can be hand drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

### 6.3 Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

None -> terminal only, so no windows/gui associated with system

## 7. REQUIREMENTS MATRIX

Provide a cross reference that traces components and data structures to the requirements in your SRS document.

Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.

## 8. APPENDICES

*This section is optional.*

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.