District Modifier: System Requirements Specification

October 19, 2017

Hyreus LTD.

Austin DeLauney Dylan Demchuk Brad Harmening

Ben Kolarik Damian Overton Jamal Savoy

Customer: Richard Chang

Contents

1 Introduction

1.1 Purpose of This Document

The purpose of this Systems Requirements Specification document is to be used as a guideline throughout the development of the application. It will list the functional and non-functional requirements and their details. There will be a contain an User Interface section that details the front-end creation of the application. Each deliverable will be listed along with a due date for its submission. Lastly, open issues will be listed that will be further investigated and discussed the further the development process goes. It is intended that our System Requirement Specification informs the customer of the goals we intend to accomplish and the dates we will accomplish them. This document is also relevant to our company, as it will be used to keep us on schedule and to remind us of the main requirements of our application.

1.2 References

QGIS Development Team, QJIS, (2002), GitHub repository, https://github.com/qgis/QGIS

United States Census Bureau. (2010). *Maryland Population and Housing Unit Counts - Blocks* [Data File]. Retrieved from https://www.census.gov/geo/maps-data/data/tiger-data.html

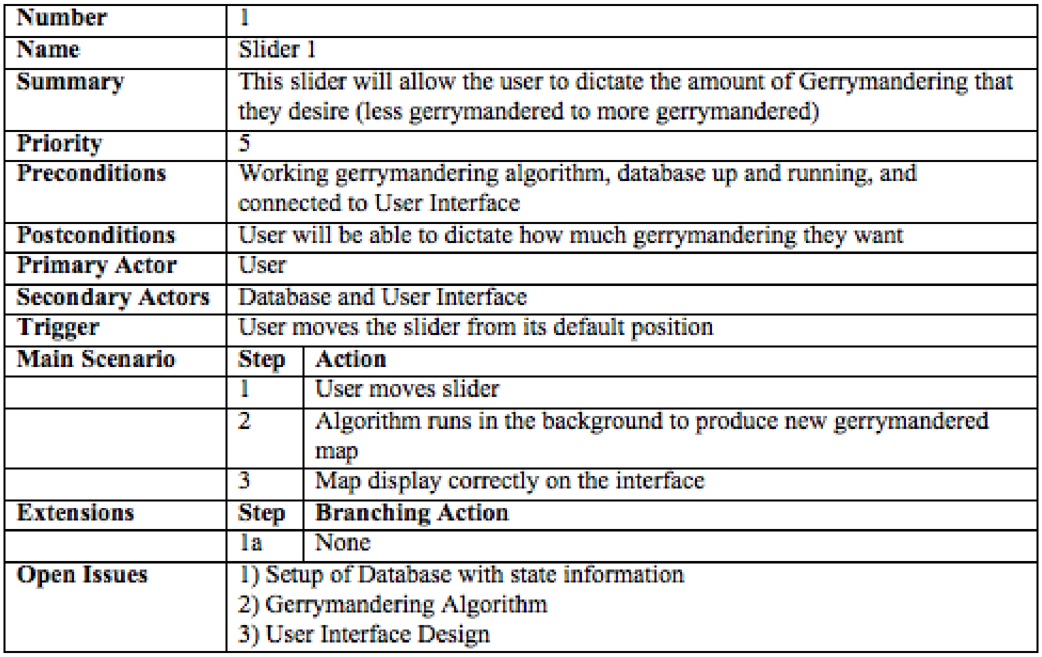
1.3 Purpose of the Product

Since the founding of the United States, gerrymandering has been used in order to strengthen one party's chance at winning an election. By splitting up voting districts strategically, parties skew the political landscape of their specific state, giving them an advantage. In a hypothetical situation, the districts would be divided in a way that portrays the actual data collected on registered voters. This program will provide the user with a way to see this hypothetical situation as well as create their own imbalances based on the percentages of party dominance they desire.

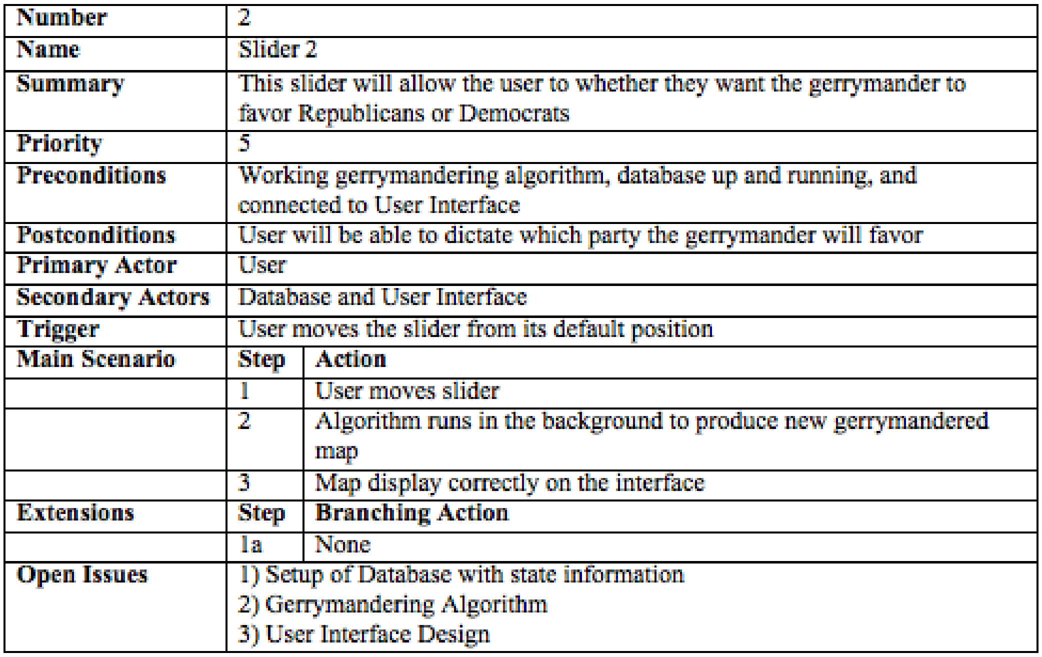
1.4 Product Scope

This application will be connected to a database that will hold voter information on the state of Maryland. The database will be created to allow room for more states to be added, but we will be specifically be focusing on the state of Maryland. Using research on gerrymandering, we will use Python coding to not only to efficiently gerrymander a state, but to display it on Google Earth. Once displayed, the user will be given the option to use three sliders that will greatly effect the gerrymander display. The first slider will decide whether the user wants the map to be more or less gerrymandered. The second slider will decide whether you want the gerrymander to favor republicans or democrats. The last slider will allow the user to decide how many districts they desire on their custom map (range is between 10-100).

2 Functional Requirements



Number	1	
Name	Slider 1	
Summary	This slider will allow the user to dictate the amount of Gerrymandering that they desire (less gerrymandered to more gerrymandered)	
Priority	5	
Preconditions	Working gerrymandering algorithm, database up and running, and connected to User Interface	
Postconditions	User will be able to dictate how much gerrymandering they want	
Primary Actor	User	
Secondary Actors	Database and User Interface	
Trigger	User moves the slider from its default position	
Main Scenario	**Step**	**Action**
	1	User moves slider
	2	Algorithm runs in the background to produce new gerrymandered map
	3	Map display correctly on the interface
Extensions	**Step**	**Branching Action**
	1a	None
Open Issues	1) Setup of Database with state information 2) Gerrymandering Algorithm 3) User Interface Design	



Number	2	
Name	Slider 2	
Summary	This slider will allow the user to whether they want the gerrymander to favor Republicans or Democrats	
Priority	5	
Preconditions	Working gerrymandering algorithm, database up and running, and connected to User Interface	
Postconditions	User will be able to dictate which party the gerrymander will favor	
Primary Actor	User	
Secondary Actors	Database and User Interface	
Trigger	User moves the slider from its default position	
Main Scenario	**Step**	**Action**
	1	User moves slider
	2	Algorithm runs in the background to produce new gerrymandered map
	3	Map display correctly on the interface
Extensions	**Step**	**Branching Action**
	1a	None
Open Issues	1) Setup of Database with state information 2) Gerrymandering Algorithm 3) User Interface Design	

3 Non-Functional Requirements

1. Response Time : The application should be able to give a response to the user's input within 400-500ms. The NFR rating is a 5. We will test this by running our animation in concert with a python clock packages to test how fast the algorithm runs as well as how fast the display appears.

2. Effectiveness : The performance of our algorithms will produce accurate results in a timely matter. The results will be displayed correctly on the application with the use of sliders. There will be extensive research done on the best algorithms, and with some changes we will come up with the best way to decide how to gerrymander a region. The NFR rating is a 5.

3. Documentation : With each paper requirement needed, it will be reviewed by the team and the customer before being pushed to our shared GitHub repository. Once pushed to the repository, we will submit the documents officially on blackboard as well as emailing the documents to the customer. The NFR rating is a 4.

4. Reliability : The application should to withstand rapid changes to the sliders by the users while being able to display the correct data

correspond to those changes. Crashes should be handled gracefully. This NFR rating is a 4. We will test this by stress testing the database as well as finding beta users to test the user interface

5. Scalability : While our main focus is the state of Maryland, there should be an ability to add more states and be able to run the same algorithms on those added areas. The NFR rating is a 3. We will test this by testing the database with additional states.
6. Usability : The application will be not only efficient as far as showing the user how to effectively gerrymander a state, but also be educational. User's will find it easy to use the sliders and understand what they are affecting. The NFR rating is a 3. We will test this by doing beta testing and using the feedback to better our project.
7. Transparency : There will be an open dialogue weekly with the customer where problems are discussed as well as progress made on whichever action has been assigned. The NFR rating is a 3. If the point of contact for our group can't physically make the meeting, an email will be sent with all of the information that would be discussed in the "in-person" meeting.
8. Quality : Code written should lead to no fatal errors. Will be written under camelCase coding standards, as well as using tabs. The NFR rating is a 2.
9. Readability : While the code will be written by six people, it will look seamless as if 1 person wrote it. There will be consistent coding standards throughout as well as useful comments. The NFR rating is a 2.
10. Security : In our case security will not be as big of a focus, since there will be no personal information released, as well as this is a very small scale project. The NFR rating is a 1.

4 User Interface

See the User Interface Design Document for the Gerrymandering Application

5 Deliverables

Hard copies of each of the following:

- Systems Requirement Specification (Due October 19)
- System Design Document (Due October 26)

- User Interface Design Document (Due October 26)
- User Manual (Due November 30)
- Administrator Manual (Due December 5)
- Copies of all Biweekly Status Reports (Bi-Weekly)

An electronic file containing the following:

- Systems Requirement Specification (Due October 19)
- System Design Document (Due October 26)
- User Interface Design Document (Due October 26)
- User Manual (Due November 30)
- Administrator Manual (Due December 5)
- All source code (Due December 12)
- The executable program (Due December 12)
- Any other software required for installation and execution of the delivered program (Due December 12)

6 Open Issues

1. The specific details on how the gerrymandering algorithm. We're still deciding the specific equations and desired runtime for the algorithm. We project this will be resolved by October 26.
2. The total collection and analysis of the voter census data. We project this will be resolved by October 26.
3. Specifics on the aesthetics of the user interface. We project this will be resolved by October 26.
4. Whether or not to have a backup for the database, and if so how that will be implemented. We project this will be resolved by October 26.
5. The number of fields for each table in the database. We project this will be resolved by October 26.

7 Appendix A - Agreement Between Customer and Contractor

Check attached document.

8 Appendix B - Team Review Sign-Off

Check attached document.

9 Appendix C - Document Contributions

Check attached document.