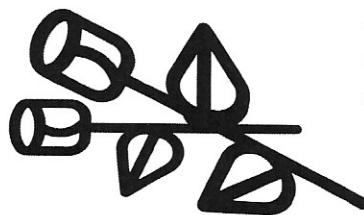


---

# District Modifier: System Requirements Specification

OCTOBER 19, 2017



# HyreUS LTD.

AUSTIN DELAUNAY   DYLAN DEMCHUK   BRAD HARMENING

BEN KOLARIK   DAMIAN OVERTON   JAMAL SAVOY

CUSTOMER: RICHARD CHANG

## **Contents**

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose of This Document . . . . .	1
1.2	References . . . . .	1
1.3	Purpose of the Product . . . . .	1
1.4	Product Scope . . . . .	1
<b>2</b>	<b>Functional Requirements</b>	<b>4</b>
<b>3</b>	<b>Non-Functional Requirements</b>	<b>7</b>
<b>4</b>	<b>User Interface</b>	<b>8</b>
<b>5</b>	<b>Deliverables</b>	<b>8</b>
<b>6</b>	<b>Open Issues</b>	<b>9</b>
<b>7</b>	<b>Appendix A - Agreement Between Customer and Contractor</b>	<b>9</b>
<b>8</b>	<b>Appendix B - Team Review Sign-Off</b>	<b>9</b>
<b>9</b>	<b>Appendix C - Document Contributions</b>	<b>10</b>

# **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**

<http://bdistricting.com/2010/> <http://www.qgis.org/en/site/>

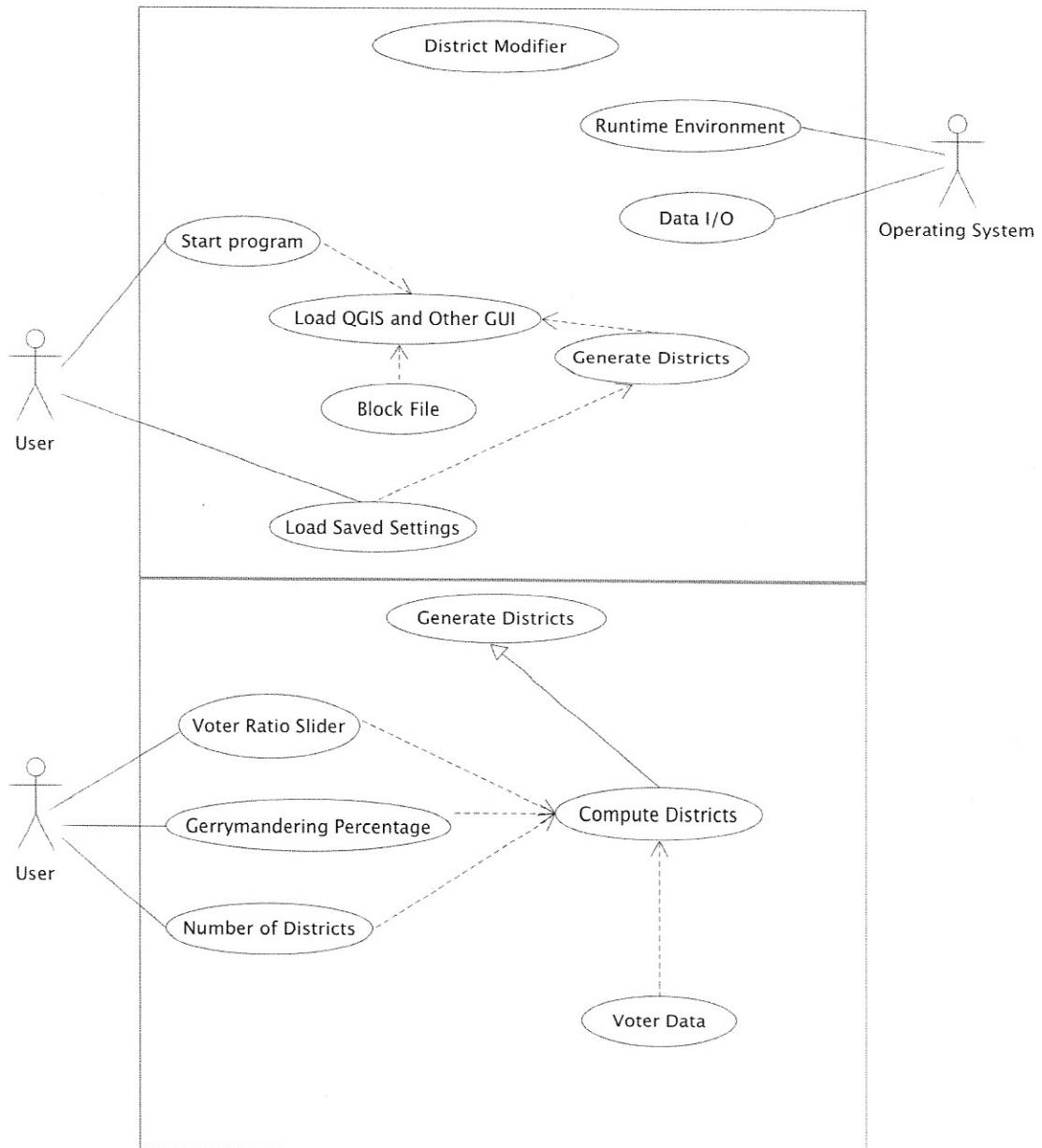
## **1.3 Purpose of the Product**

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 QGIS. 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

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

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

<b>Number</b>	3	
<b>Name</b>	Adjusting number of districts (between 10-100)	
<b>Summary</b>	This slider will allow the user to choose how many districts they want displayed on the map	
<b>Priority</b>	5	
<b>Preconditions</b>	Working gerrymandering algorithm, database up and running, and connected to User Interface	
<b>Postconditions</b>	User is able to dictate number of districts	
<b>Primary Actor</b>	User	
<b>Secondary Actors</b>	Database and User Interface	
<b>Trigger</b>	User moves the slider from its default position	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User moves slider
	2	Algorithm runs in the background to produce new gerrymandered map
	3	Map display correctly on the interface
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	None
<b>Open Issues</b>	1) Setup of Database with state information 2) Gerrymandering Algorithm 3) User Interface Design	

<b>Number</b>	4	
<b>Name</b>	Working algorithm in the background	
<b>Summary</b>	The algorithm should be accurate yet efficient, producing a result within 400-500ms	
<b>Priority</b>	5	
<b>Preconditions</b>	Working gerrymandering algorithm, database up and running, and connected to User Interface	
<b>Postconditions</b>	Algorithm will return an accurate display of the gerrymandered districts	
<b>Primary Actor</b>	User	
<b>Secondary Actors</b>	Census data included in the Database	
<b>Trigger</b>	User moves the slider from its default position	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User moves slider
	2	Algorithm runs in the background to produce new gerrymandered map
	3	Algorithm returns correct results
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	Research algorithm, code in python, and test it using census data
<b>Open Issues</b>	1) Setup of Database with state information 2) Gerrymandering Algorithm 3) User Interface Design	

<b>Number</b>	5	
<b>Name</b>	Saving of slider settings	
<b>Summary</b>	Users should be able to save whichever slider settings they desire (up to 3)	
<b>Priority</b>	2	
<b>Preconditions</b>	Users find a combination of slider settings that they desire, and click save	
<b>Postconditions</b>	They will now be able to reload up to three previous slider settings	
<b>Primary Actor</b>	User	
<b>Secondary Actors</b>	The database and user interface, along with working algorithm	
<b>Trigger</b>	User attempts to save slider settings	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User finds desired settings for each slider
	2	User clicks save button on screen and names slider settings
	3	Settings are saved and now able to be reloaded on the screen
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	None
<b>Open Issues</b>	None	

<b>Number</b>	6	
<b>Name</b>	Working application through loss of internet connection	
<b>Summary</b>	Application should be fully functional without initial internet connection, and during loss of connectivity	
<b>Priority</b>	2	
<b>Preconditions</b>	Connection is lost or there was never an internet connection	
<b>Postconditions</b>	Regardless, application is still fully functional	
<b>Primary Actor</b>	User	
<b>Secondary Actors</b>	Every part of the system	
<b>Trigger</b>	User loses connectivity	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User loses connectivity or starts application without internet connectivity
	2	Application should proceed to keep running efficiently after loss of connectivity
	3	User should notice no difference in using the application
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	None
<b>Open Issues</b>	None	

<b>Number</b>	7	
<b>Name</b>	Graceful Failure of Application	
<b>Summary</b>	This slider will allow the user to whether they want the gerrymander to favor Republicans or Democrats	
<b>Priority</b>	5	
<b>Preconditions</b>	Working gerrymandering algorithm, database up and running, and connected to User Interface	
<b>Postconditions</b>	User will be able to dictate which party the gerrymander will favor	
<b>Primary Actor</b>	User	
<b>Secondary Actors</b>	Database and User Interface	
<b>Trigger</b>	User moves the slider from its default position	
<b>Main Scenario</b>	<b>Step</b>	<b>Action</b>
	1	User moves slider
	2	Algorithm runs in the background to produce new gerrymandered map
	3	Map display correctly on the interface
<b>Extensions</b>	<b>Step</b>	<b>Branching Action</b>
	1a	None
<b>Open Issues</b>	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 package 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 manner. 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 using the results of this research, 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 withstand rapid changes to the sliders by the users while being able to display the correct data

corresponding 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, the application should provide the ability to add additional 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 and application 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. The code 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 one 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 personally identifiable 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 the gerrymandering algorithm (equations, run time, etc...).
2. The collection and analysis of the voter census data.
3. Specifics on the aesthetics of the user interface.
4. Whether or not to have a backup for the database, and if so how that will be implemented.
5. The number of fields for each table in the database.

## 7 Appendix A - Agreement Between Customer and Contractor

See attached document.

## 8 Appendix B - Team Review Sign-Off

See attached document.

## **9 Appendix C - Document Contributions**

See attached document.

# System Requirement Agreement

Between

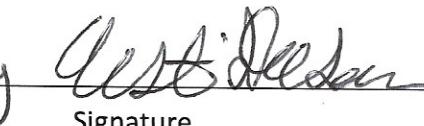
Hyreus LTD.  
University of Maryland, Baltimore County. 1000 Hilltop Circle  
Baltimore, MD 21250  
USA

and

Dr. Richard Chang  
University of Maryland, Baltimore County. 1000 Hilltop Circle  
Baltimore, MD 21250  
USA

By signing below, the customer [Dr. Richard Chang], and company [Hyreus LTD.] are agreeing to the design and process detailed in the System Requirement Specification document. We are both agreeing to the scope of the project as well as its functional and non-functional requirements. We all agree on the deliverables, their due dates, as well as how they will be presented to the customer.

In the case that changes will need to be made, the customer will contact the point of contact for our company [Austin DeLauney] with what changes he requests. Then we will discuss separately as a team on whether or not the request is accomplishable given the time left on the project. We will then have a formal meeting with the customer to formally agree or disagree to the request. The request, if agreed upon, will then be recorded on a separate document and uploaded to the shared repository.

Austin DeLauney  10/17/2017

Print Name

Signature

Date

Brad Harmening  10/19/17

Print Name

Signature

Date

Jamal Savoy  10/19/17

Print Name

Signature

Date

Dylan Demchuk  10/19/2017

Print Name

Signature

Date

Dominic Averton

Print Name

Dominic Averton

Signature

10/19/17

Date

Benjamin Kolanze

Print Name

Benjamin Kolanze

Signature

10-19-17

Date

---

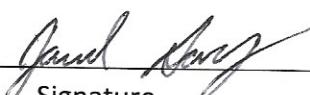
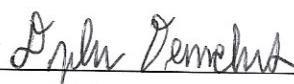
Print Name

Signature

Date

## Team Review Sign Off

The members of Hyreus LTD. have reviewed the Systems Requirements and agreed to its contents and formats. After several discussions and meetings with the customer we were able to formulate a design for our application as well as our process. Below will contain the signatures of each team member agreeing to the terms of the project's requirements. Any comments about minor disagreements that any of the members have had will be included below.

Austin DeLauney		10/19/2017
Print Name	Signature	Date
Benjamin Kolarik		10/19/2017
Print Name	Signature	Date
Brad Harmening		10/19/17
Print Name	Signature	Date
Jamal Savoy		10/19/17
Print Name	Signature	Date
Dylan Demchuk		10/19/17
Print Name	Signature	Date
Demian Overton		10/19/17
Print Name	Signature	Date

Comments:

Austin DeLauney \_\_\_\_\_

Dylan Demchuk \_\_\_\_\_

Brad Harmening \_\_\_\_\_

Ben Kolarik \_\_\_\_\_

Damian Overton \_\_\_\_\_

Jamal Savoy No comments. Agree with the design N/A