System Requirements Specification

CMSC 447

Districtlands

Customer: Russell Cain

Team: Red·de

Members:

Dorothy Carter
Joanna Dinh
Khang Ngo
Kwame Owusu-Boaitey
Taylor Brzuchalski
Stephen Lin

Districtlands
System Requirements Specification

Table of Contents

1. Introduction

1.1 Purpose of This Document

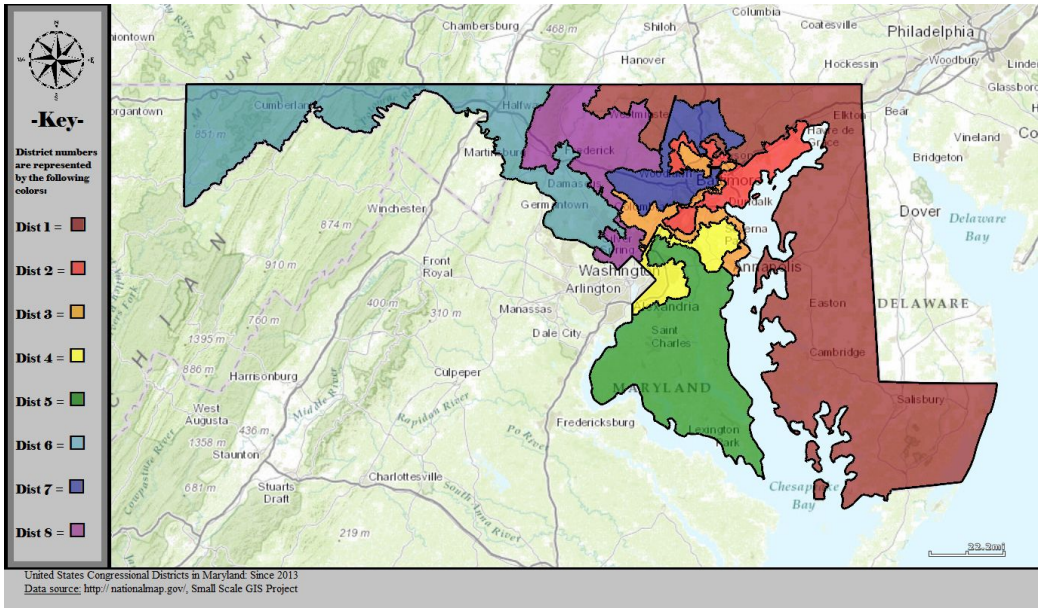
This document is designed to specify and explain the functionalities of the Districtlands web application software; its features, requirements for running, and UI experience. The intended readership are the students of Professor Terry Yoo's Software Engineer class, the client, Russell Cain, and any interested audience in redistricting Maryland in an unbiased representation.

1.2 References

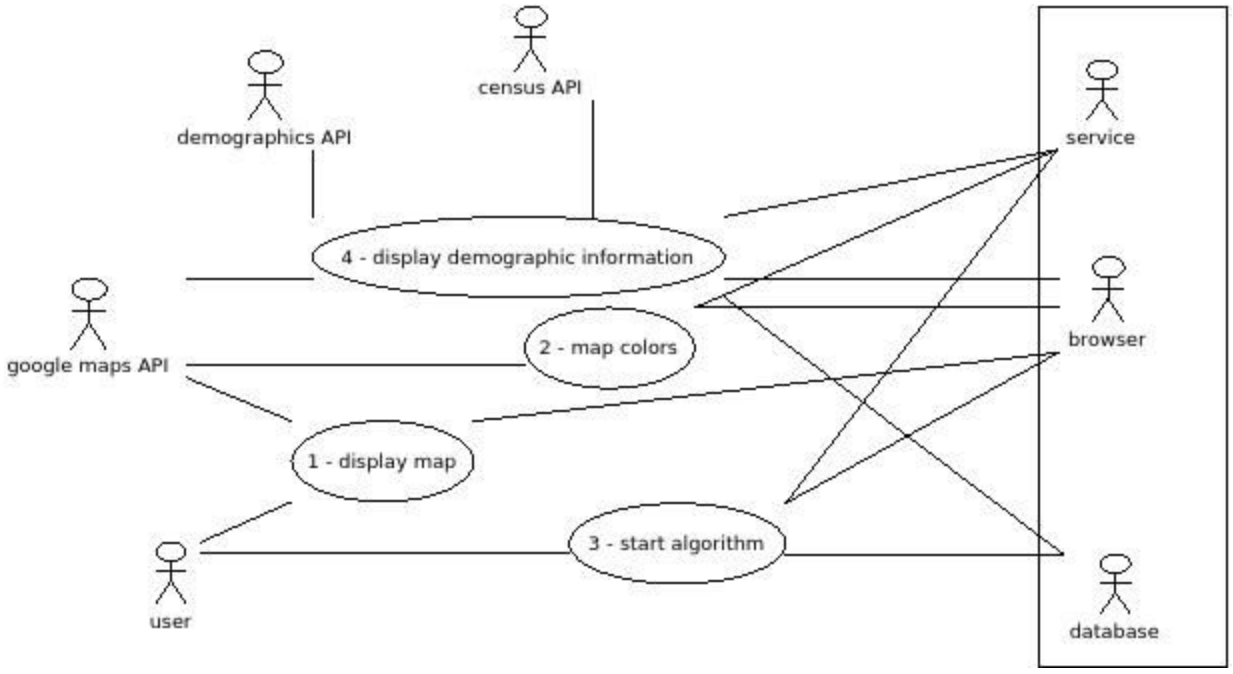
- https://developers.google.com/maps/
- https://github.com/uscensusbureau/citysdk/blob/master/src/api/README.md
- https://github.com/CommerceDataService/citysdk/blob/master/README.md
- https://mappingsupport.com/p/politics/congressional-district-map-tips.html

1.3 Purpose of the Product

Gerrymandering. Maryland has the worst gerrymandering of electoral districts in the United States. Gerrymandering is used in order to favor political parties or hinder certain demographics resulting in unequal and biased districts. This product aims to better review the districts by using an algorithm to redraw the borderlines for a more accurate, unbiased representation of the demographics, based on user specifications.



1.4 Product Scope



The product will provide a system that allows a user to redistrict Maryland in a way of his or her choosing that would make it fair and return the results. The user will be provided with a web interface with a map of Maryland. Clicking on a part of the map will trigger an algorithm to fairly redraw the districts using demographic data files we have obtained. Redrawn districts will be displayed on the map, each in a unique color, and demographic information will be displayed alongside the map.

2. Functional Requirements

Use Case 1 - Map

Number	1	
Name	Load Map	
Summary	Display map of Maryland	
Priority	5	
Preconditions	Browser can support the map	
Postconditions	Map of Maryland is displayed by browser	
Primary Actor	User	
Secondary Actors	Google Map API, web server, web application	
Trigger	User runs the web application or enters the URL of the web application	
Main Scenario	**Step**	**Action**
	1	Enter URL of web application in browser
	2	Web page loads map of Maryland
Extensions	**Step**	**Branching Action**
	2a	Person refreshes browser
Open Issues	Web page should be loadable by different browsers	

Use Case 2 - Colors

Number	2	
Name	Display Colors	
Summary	Each district should be uniquely color coded	
Priority	4	
Preconditions	The districts have been redrawn	
Postconditions	New districts are uniquely color coded	
Primary Actor	User	
Secondary Actors	Google Map API, Backend services, Javascript program	
Trigger	User inputs specification to run algorithm to redraw districts	
Main Scenario	**Step**	**Action**
	1	User clicks specification of how to redraw districts
	2	Algorithm runs best points for new districts
	3	Google Maps API plots and color codes new districts
Extensions	**Step**	**Branching Action**
	1a	Person changes specification and runs algorithm again
Open Issues	Avoid having the same color district next to each other	

Use Case 3 - Start Algorithm

Number	3	
Name	Gerrymandering Start	
Summary	The system shall perform the gerrymandering algorithm and display the results upon the user's action	
Priority	3	
Preconditions	The map of Maryland is loaded, the user clicks a button to begin the algorithm from a predetermined location, decides to personally select a position to begin the algorithm from	
Postconditions	The map displays districted regions calculated by the algorithm	
Primary Actor	User	
Secondary Actors	Google Map API, backend services, web application, database server	
Trigger	User uses one of the methods available to begin the gerrymandering operation	
Main Scenario	**Step**	**Action**
	1	User clicks button to begin algorithm from predetermined position
	2	Algorithm runs best points for new districts
	3	Google Maps API plots and color codes new districts
	4	User interacts with map to view results
Extensions	**Step**	**Branching Action**
	1a	User selects option to run algorithm from user-decided position
	2a	User selects point on map by clicking
	3a	Algorithm begins calculating with given coordinate as a starting location
	3	(Resume main scenario) Google Maps API plots and color codes new districts
Open Issues	-How a given start coordinate should affect algorithm operation is not yet fully known	

Use Case 4 - Demographics

Number	4	
Name	Display Demographics	
Summary	The system shall provide demographic and other related information upon the user's request.	
Priority	2	
Preconditions	1. The map of Maryland is loaded, the user has performed an action related to the demographic functionality of the program 2. The map of Maryland is loaded, the gerrymandering algorithm has run and districts are shown on the map, the user has clicked on a shaded district	
Postconditions	The program displays general demographic information or information pertaining to some kind of request inputted by the user when they initially activated the function	
Primary Actor	User	
Secondary Actors	Google Map API, backend services, web application, database server, other data related APIs	
Trigger	User uses one of the demographic features available	
Main Scenario (Precondition 1 and 2)	**Step**	**Action**
	1	User clicks a button to display general demographic information
	2	The program displays general demographic information and other possible information
Extensions (Precondition 2)	**Step**	**Branching Action**
	1a	User clicks a district on the map
	2a	Program displays information related to the clicked district
Open Issues	-It is not currently known if it is possible to get demographic information currently -It is not yet known how to determine the demographic statistics for a region just created by the program, given access to general demographic information -The specific information we would like to display has not been decided	

2a. Functional Requirements

Requirement 1 - Map
The Districtlands tool shall display the map of Maryland upon launch of the web application.

Requirement 2 - Colors
The Districtlands tool shall uniquely color code each new district drawn.

Requirement 3 - Algorithm specification
The Districtlands tool shall allow the user to specify how the algorithm redraws the district lines.

Requirement 3a - Algorithm Start (Predetermined position)
The algorithm shall begin from a predetermined location.

Requirement 3b - Algorithm Start (User-determined position)
The algorithm shall begin at a user-selected location.

Requirement 3c - Algorithm Result Storage
The resulting output of the algorithm shall be stored.

Requirement 4 - Demographics (User-determined district)
Demographic information shall be displayed as a table for the district the user has selected.

Requirement 4a - Demographics (General display)
Demographic information shall be displayed as a table for all districts.

Requirement 5 - Reusability
The Districtlands tool shall be able to utilise multiple sets of demographic data.

3. Non-Functional Requirements

#	Item	Priority 1 (lowest) to 5 (highest)
1	Algorithm will make use of demographic information to perform districting actions	4
2	Web application shall be supported by the popular browsers: Chrome, Firefox.	4
3	Code shall be documented	5
4	Web Application shall be operated with JavaScript, HTML, and CSS	4
5	Backend services shall be written in Python	4
6	User shall be able to interact with displayed map by clicking directly on it.	4
7	If it is necessary for high volumes of data to be stored and utilised, databases shall be used to more efficiently access and organise data	1
8	Algorithm shall complete in a reasonable amount of time (<5s)	1
9	Application shall use a reasonable amount of system resources (<500,000kb)	1
10	Web application shall be supported on the latest versions of operating systems Windows and macOS	1

3a. Non-Functional Requirements

Non-Functional Requirement 1 - Before and After
The map shall display the voting demographics before the algorithm has been specified and then change to reflect the new voting demographics after the algorithm has been specified.

Non-Functional Requirement 2 - Web Browser
The web application's functionality shall be tested by using the main browsers listed: Safari, Chrome, Firefox.

Non-Functional Requirement 3 - Documentation
The web application shall be well documented. Source code shall also be well written and commented. Documentation shall include the Systems Requirement Specification, System Design Document, User Interface Design Document, and Administrator Manual. Members of the group shall code-check other members' code to ensure an acceptable level of understanding can be obtained from commented code.

Non-Functional Requirement 4 - JavaScript, HTML, CSS
The web application shall be written using JavaScript, HTML, and CSS.

Non-Functional Requirement 5 - Backend in Python
The backend functionality of the web application shall be written in the programming language Python.

Non-Functional Requirement 6 - Interactive Map
Interactive functions of the map (scrolling, zooming, clicking, any UI buttons, etc.) shall be usable and operate smoothly with acceptable response times by testing each UI element.

Non-functional Requirement 7 - Database Usage
The database server shall be operational and more effective than not using a database server.

Non-functional Requirement 8 - Runtime
The runtime shall be timed using the redistrict algorithm at various points.

Non-functional Requirement 9 - System Resources
The application shall be tested on its usage of system resources, using system monitoring tools to monitor resource use during application run. The application should be tested with average and large data sets.

Non-functional Test 10 - Platform Compatibility
The application shall be run and tested on systems running the latest versions of Windows and macOS.

4. User Interface

See “User Interface Design Document for Districtlands.”

5. Deliverables

Hard copies of each of the following:

- Systems Requirement Specification
- System Design Document
- User Interface Design Document
- Administrator Manual
- Copies of all Biweekly Status Reports

An electronic file containing the following:

- Systems Requirement Specification
- System Design Document
- User Interface Design Document
- Administrator Manual
- All source code
- The executable program
- Any other software required for installation and execution of the delivered program.

6. Open Issues

#	Open Issue	Priority
1	The web page should operate identically in different browser environments	2
2	The data needed for the program needs to be determined and finalized.	5
3	The district cannot be next to another district of the same color to avoid confusion between the districts.	4
4	It needs to be known how a given start coordinate will affect the algorithm operation	3
5	Information to be displayed needs to be decided on	2
6	What demographic information we have access to needs to be looked into	2
7	How to determine demographic statistics for a region produced by the program needs to be looked into	2

Appendix A – Agreement Between Customer and Contractor

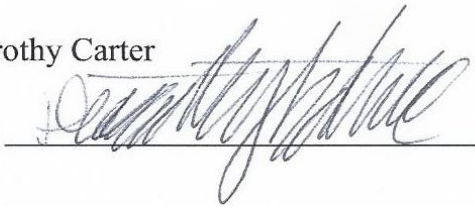
By signing this document, all parties agree that the given set of requirements are acceptable and do not need to be changed at this time.

In the case of future changes to this document, all changes must be agreed upon by the members of the team and the customer. There will be a new signing of the document to document the agreement of the changes.

Name: Russell Cain

Signature:  **Date:** 19 Oct 2017

Name: Dorothy Carter

Signature:  **Date:** 19 Oct 17

Name: Joanna Dinh

Signature:  **Date:** 10/19/2017

Name: Khang Ngo

Signature: 

Date: 10/19/2017

Name: Kwame Owusu-Boaitey

Signature: _____ **Date:** 10/19/2017

Name: Taylor Brzuchalski

Signature: _____ **Date:** 10/19/17

Name: Stephen Lin

Signature: _____ **Date:** 10/19/17

Appendix B – Team Review Sign-off

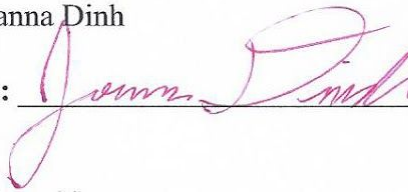
By signing this section, all members are agreeing that they have reviewed this document and agree on its content and format.

Name: Dorothy Carter

Signature:  **Date:** 19 Oct 17

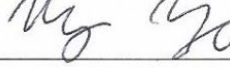
Comments:

Name: Joanna Dinh

Signature:  **Date:** 10/19/2017

Comments:

Name: Khang Ngo

Signature: 

Date: 10/19/2017

Comments:

Name: Kwame Owusu-Boaitey

Signature: _____ **Date:** 10/19/2017

Comments:

Name: Taylor Brzuchalski

Signature:  **Date:** 10/19/17

Comments:

Name: Stephen Lin

Signature:  **Date:** 10/19/17

Comments:

Appendix C – Document Contributions

Name	Section(s)/Contributions	Percentage of work
Dorothy Carter	Product scope diagram, some editing, few Test NFRs	16%
Joanna Dinh	1.1, 1.2, 1.3, few FRs/Test FRs cases, few NFRs/Test NFRs cases	19%
Khang Ngo	Added FR/NFRs+FR/NFR tests, formatting, logos	17%
Kwame Owusu-Boaitey	Added few functional test edits, proposed some team names	15%
Taylor Brzuchalski	Open Issues section, Initial document set up, Appendices, Few test cases, Some editing	17%
Stephen Lin	Product scope summary, NFR and tests, format	16%