

System Design Document

Districtlands

Customer: Russell Cain

Team: Red.de

Members:

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

Date: October 23, 2017



Districtlands
System Design Document

Table of Contents

	<u>Page</u>
1. Introduction	
1.1 Purpose of This Document	3
1.2 References	3
2. System Architecture	
2.1 Architectural Design	5
2.2 Decomposition Description	6
3. Persistent Data Design	
3.1 Database Descriptions	8
3.2 File Descriptions	8
4. Requirements Matrix	10
Appendix A – Agreement Between Customer and Contractor	11
Appendix B – Peer Review Sign-off	12
Appendix C – Document Contributions	13

1. Introduction

1.1 Purpose of This Document

This document describes the design of the Distictlands system at an architectural level. This description will allow development to proceed with details on what is to be implemented and how it will be implemented. The intended readership are the students of Doctor Terry Yoo's Software Engineering class, the customer, Russell Cain, and any interested audience in redistricting Maryland in an unbiased representation.

1.2 References

System Requirements Specification Document

<https://www.csee.umbc.edu/~tsyoo/cmsc447-Fall2017-sec3/> - Lecture slides

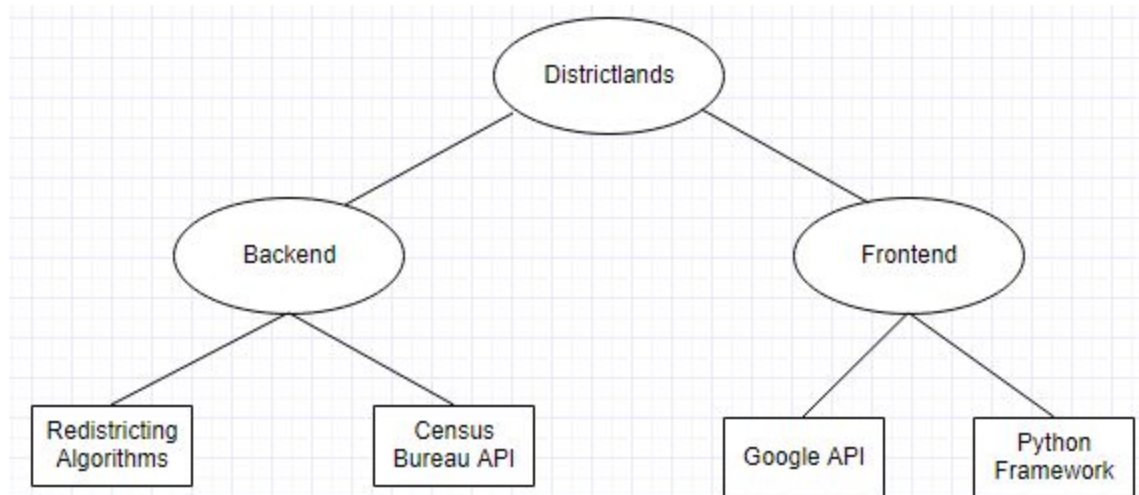
https://www.census.gov/geo/maps-data/data/kml/kml_tracts.html

<https://www.census.gov/data/developers/data-sets/decennial-census.html>

<https://api.census.gov/data/2010/sf1/variables.html>

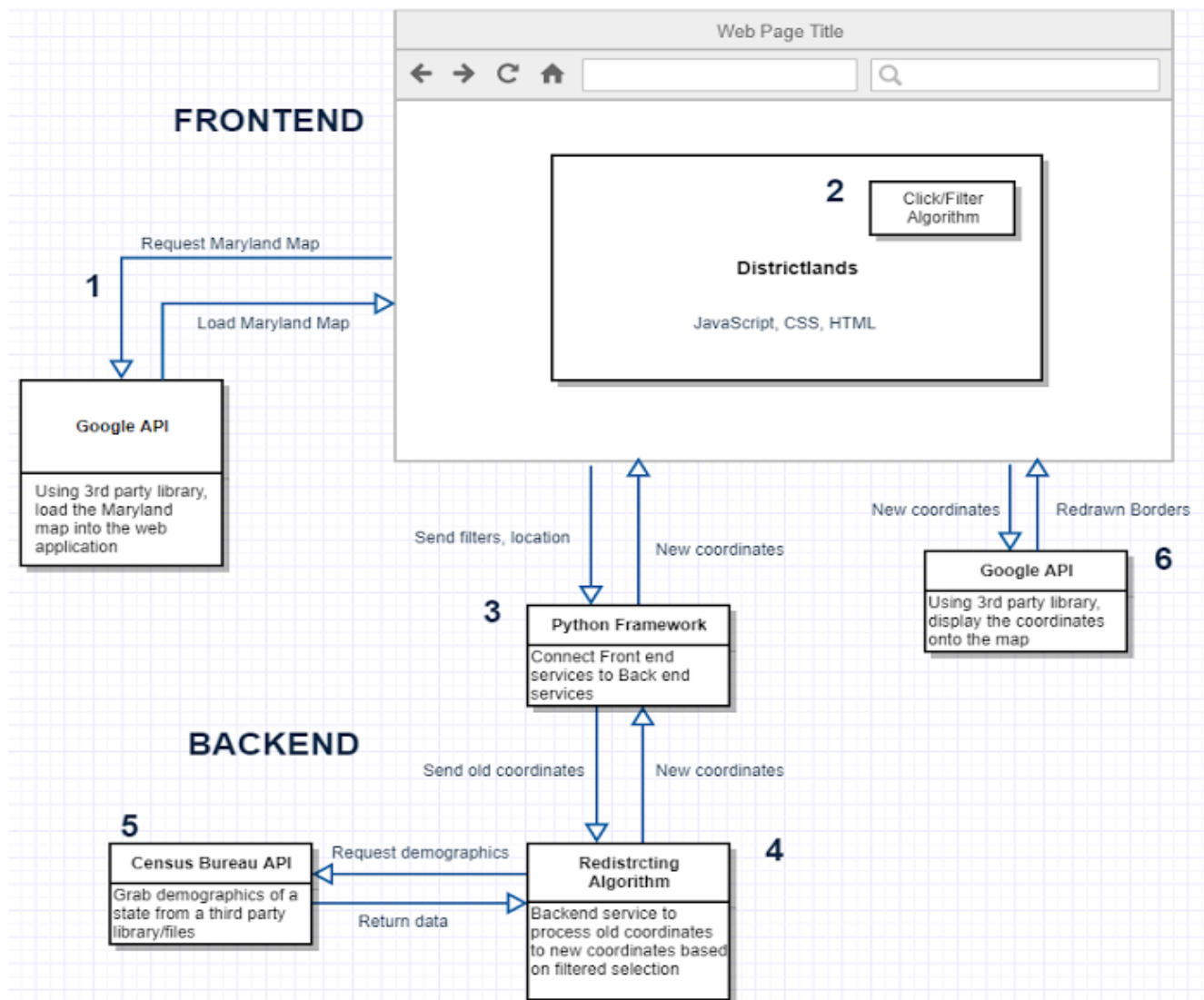
2. System Architecture

2.1 Architectural Design



Districtlands will be a web application run on the local machine. The user will interact with the frontend in the browser of their choice, and the frontend will connect via a Python framework to the backend for: data retrieval, running the algorithm, and other tasks that involve intense processing. The backend will consist of the redistricting algorithm, functions to retrieve demographic information for use both by the redistricting algorithm and by the frontend, functions to retrieve the various geographic pieces the algorithm will work with (e.g. census blocks, finished districts, etc.), and objects to represent those geographic pieces. The frontend will consist of the map, functions to display demographic data, and connections to the backend. The frontend will use the Google Maps API for its mapping tasks. The backend will use the APIs provided by the Census Bureau to retrieve population data.

2.2 Decomposition Description



1. This is the initial communication between the software, Distractlands tools, and the user upon launch of the application. Districtlands tool will grab a map of Maryland and display it using Google's API.
2. The user will select filtering options before hard clicking on the map to start the districting algorithm.
3. Wherever the cursor has clicked, the location (longitude, latitude) of that click will be sent to the backend services along with the filters via HTTP request to the Python framework (Django) as the starting point of the redistricting algorithm where the heavy processing occurs. New coordinates from the back-end will be sent back to the front-end to be processed into the map.
4. Given a starting location point and filtering requirements, the redistricting algorithm function will pull data from the Census Bureau API to determine the next best plot points based on the given requirements. The function will return new coordinates as a HTTP response.

5. The Census Bureau API is a third party library used to request and collect data used by the Redistricting Algorithm.
6. The front-end will process any new coordinates onto the map using Google's API, closing the communication between Districtlands and the user.

3. Persistent Data Design

3.1 Database Descriptions

No database will be used.

3.2 File Descriptions

Files used by Districtlands all come from the various APIs that it uses, and none should have to be created by a programmer or user. Most of these are in JSON, but the ones defining the census tract borders are KML. Samples of both have been included below.

3.2.1 Census Demographics JSON Example

```
[["P0010001", "P0030001", "NAME", "state", "county", "tract"],
["3718", "3718", "Census Tract 1", "24", "001", "000100"],
["4564", "4564", "Census Tract 2", "24", "001", "000200"],
["2780", "2780", "Census Tract 3", "24", "001", "000300"],
["3022", "3022", "Census Tract 4", "24", "001", "000400"],
["2734", "2734", "Census Tract 5", "24", "001", "000500"],
["2965", "2965", "Census Tract 6", "24", "001", "000600"],
["3387", "3387", "Census Tract 7", "24", "001", "000700"],
...
["4101", "4101", "Census Tract 2803.01", "24", "510", "280301"],
["2282", "2282", "Census Tract 2803.02", "24", "510", "280302"],
["3565", "3565", "Census Tract 2804.01", "24", "510", "280401"],
["1574", "1574", "Census Tract 2804.02", "24", "510", "280402"],
["5073", "5073", "Census Tract 2804.03", "24", "510", "280403"],
["2267", "2267", "Census Tract 2804.04", "24", "510", "280404"],
["3549", "3549", "Census Tract 2805", "24", "510", "280500"]]
```

3.2.2 Census KML Example

```
<Placemark id="cb_2016_24_tract_500k.kml">
  <name>&lt;at&gt;&lt;openparen&gt;24005441101&lt;closeparen&gt;</name>
  <visibility>1</visibility>
  <description>
    <![CDATA[<center><table><tr><th colspan='2'
      align='center'><em>Attributes</em></th></tr><tr
      bgcolor="#E3E3F3">
        <th>STATEFP</th>
        <td>24</td>
      </tr><tr bgcolor="">
        <th>COUNTYFP</th>
        <td>005</td>
      </tr><tr bgcolor="#E3E3F3">
        <th>TRACTCE</th>
        <td>441101</td>
      </tr><tr bgcolor="">
        <th>AFFGEOID</th>
        <td>1400000US24005441101</td>
      </tr><tr bgcolor="#E3E3F3">
```

```

        <th>GEOID</th>
        <td>24005441101</td>
      </tr><tr bgcolor="">
        <th>NAME</th>
        <td>4411.01</td>
      </tr><tr bgcolor="#E3E3F3">
        <th>LSAD</th>
        <td>CT</td>
      </tr><tr bgcolor="">
        <th>ALAND</th>
        <td>1530290</td>
      </tr><tr bgcolor="#E3E3F3">
        <th>AWATER</th>
        <td>0</td>
      </tr></table></center>]]>
</description>
<styleUrl>#KMLStyler</styleUrl>
<ExtendedData>
  <SchemaData schemaUrl="#kml_schema_ft_cb_2016_24_tract_500k">
    <SimpleData name="STATEFP">24</SimpleData>
    <SimpleData name="COUNTYFP">005</SimpleData>
    <SimpleData name="TRACTCE">441101</SimpleData>
    <SimpleData
      name="AFFGEOID">1400000US24005441101</SimpleData>
    <SimpleData name="GEOID">24005441101</SimpleData>
    <SimpleData name="NAME">4411.01</SimpleData>
    <SimpleData name="LSAD">CT</SimpleData>
    <SimpleData name="ALAND">1530290</SimpleData>
    <SimpleData name="AWATER">0</SimpleData>
  </SchemaData>
</ExtendedData>
<Polygon>
  <extrude>0</extrude>
  <tessellate>1</tessellate>
  <altitudeMode>clampToGround</altitudeMode>
  <outerBoundaryIs>
    <LinearRing>
      <coordinates>-76.514165,39.340075,0.0
        -76.514759,39.33851,0.0 -76.512792,39.33519,0.0
        -76.5115,39.331768,0.0 -76.510193,39.331134,0.0
        -76.509985,39.328306,0.0 -76.513102,39.325364,0.0
        -76.512934,39.323344,0.0 -76.514656,39.320787,0.0
        -76.513922,39.321272,0.0 -76.495709,39.333535,0.0
        -76.499759,39.335858,0.0 -76.501163,39.336536,0.0
        -76.50947,39.339097,0.0 -76.514035,39.341454,0.0
        -76.514707,39.340256,0.0
        -76.514165,39.340075,0.0</coordinates>
    </LinearRing>
  </outerBoundaryIs>
</Polygon>
</Placemark>

```


4. Requirements Matrix

req #	Requirement	Component/module	Description
1	Map	Webpage UI, Webpage Map Rendering	The Webpage will handle displaying of the map and UI elements, while the component interacting with Google Maps API will provide additional map functionality
2	Colors	Webpage Map Rendering (or) Backend Redistricting	The frontend based Map Rendering module will interact with Google Maps API to colour the output. Alternatively, the redistricting algorithm may instead perform this color determination
3	Algorithm	Backend Redistricting Program	The districting algorithm of the program performs districting operations
3a	Algorithm Start, Predetermined	Webpage UI	The UI component will register a user's intent to start from a predetermined location
3b	Algorithm Start, User Determined	Webpage UI, Data Processing	The UI component will register a user input, the input will be converted as necessary with the Data Processing module to work with the backend algorithm
3c	Results Storage	Outputting, Data Processing	An outputting component will be able to output data in a format re-loadable by the program. The data will be processes with the Data Processing module.
4	Demographics (district)	Data Processing, Census Data API Interfaces, Webpage UI, District Data Processing	Information will be retrieved with the Data Interfaces module of the program, processed if necessary with Data Processing, and potentially re-processed and divided with the District Data Processing module. The data will then be displayed with the UI.
4a	Demographics (General)	Data Processing, Census Data API Interfaces, Webpage UI	Information will be retrieved with the Data Interfaces module of the program, processed if necessary with Data Processing, and displayed with the Webpage UI.

5	Reusability	Census Data API Interfaces	The part of the program interacting with various Data APIs will be able to access and handle data from different sources.
---	-------------	----------------------------	---------------------------------------------------------------------------------------------------------------------------

Appendix A – Agreement Between Customer and Contractor

By signing off on this document, all parties agree that the information presented in this document is what is required and is acceptable.

In case of future changes, all changes must be agreed upon by all parties. This agreement will be documented with a resigning of this document.

Name: Russell Cain

Signature:  Date: 24 Oct 2017

Comments:


Name: Dorothy Carter

Signature:  Date: 24 Oct 17

Name: Joanna Dinh

Signature:  Date: 10-24-17


Name: Khang Ngo

Signature:  Date: 10/24

Name: Kwame Owusu-Boateng

Signature:  Date: 10/24

Name: Taylor Brzuchalski

Signature:  Date: 10/24/17


Name: Stephen Lin

Signature:  Date: 10/24/17

Appendix B – Team Review Sign-off

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

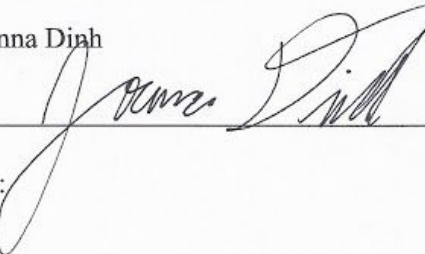
Name: Dorothy Carter

Signature: 

Date: 24 Oct 17

Comments:

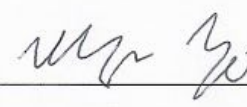
Name: Joanna Dinh

Signature: 

Date: 10-24-17

Comments:

Name: Khang Ngo

Signature: 

Date: 10/24

Comments:


Name: Kwame Owusu-Boateng

Signature: 

Date: 10/24

Comments:

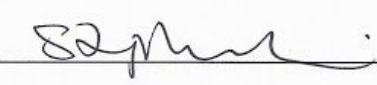
Name: Taylor Brzuchalski

Signature: 

Date: 10/24/17

Comments:

Name: Stephen Lin

Signature: 

Date: 10/24/17

Appendix C – Document Contributions

Name	Section(s)/Contributions	Percentage of work
Dorothy Carter	2.1, 3.2	16
Joanna Dinh	2.2	16
Khang Ngo	Requirements matrix	18
Kwame Owusu-Boaitey	2.1's UML, small edits	16
Taylor Brzuchalski	1.1, 1.2, 3.1, Appendices, Formatting	18
Stephen Lin	Formatting and final edits	16