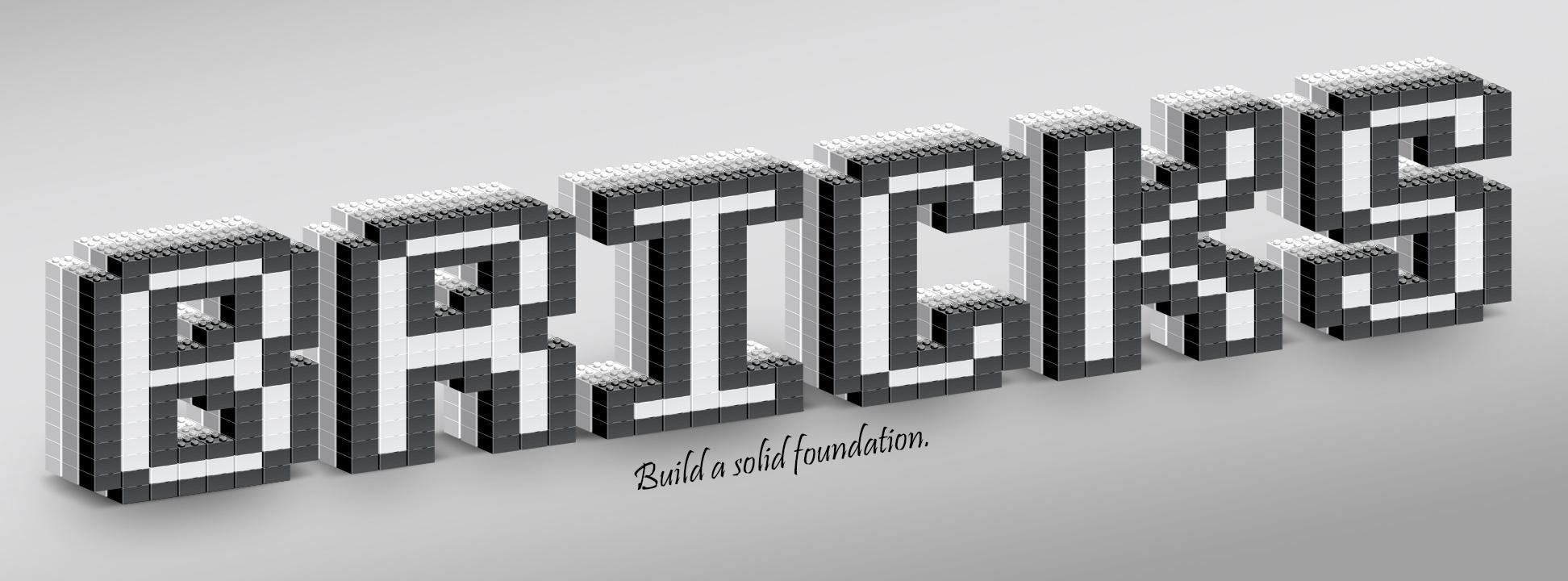
CMSC 447: Software Engineering I December 5, 2017

# **Administrator Manual**

# **The Redistrictinator**

Client: Geoffrey Weiss



**BRICKS**

Benjamin Jeremenko

Jacob Philip

Sumanth Neerumalla

Zachary Elliott

Francis Kato

Nathaniel Fuller

**The Redistrictinator**

**Administrator Manual**

**Table of Contents**

**Page**

**1. Introduction 3**

**1.1 Purpose of This Document 3**

**1.2 References 3**

**2. System Overview 4**

**2.1 Background 4**

**2.2 Hardware and Software Requirements 4**

**3. Administrative Procedures 5-7**

**3.1 Installation 5-6**

**3.2 Routine Tasks 6**

**3.3 Backups 6**

**3.4 User Support 7**

**4. Troubleshooting 7-8**

**4.2 Dealing with Error Messages and Failures 7**

**4.3 Known Bugs and Limitations 8**

**Appendix A – Peer Review Sign-off 9**

**Appendix B – Document Contributions 10**

**1. Introduction**

The purpose of this document is to serve as a reference and users guide for any individual that wishes to run, use, and maintain the Redistrictinator software.

**1.1 Purpose of This Document**

This document contains comprehensive information on the hardware/software requirements, the installation process, maintenance requirements, how to support users , dealing with errors, as well as any known shortcomings of the software product.

The software has been designed to take up minimal space and have minimal requirements for operation. If installed properly and used within the planned use cases, it should be useful to users for the foreseeable long term future.

**1.2 References**

A variety of documents and information sources were used to create this document and software.

* *System Requirements Specification Document,* The Redistrictinator
* *System Design Document,* The Redistrictinator
* AWS EC2 Documentation, Retrieved 12/5/2017, from <https://aws.amazon.com/documentation/ec2/>
* Namecheap Documentation, Retrieved 12/5/2017, from <https://www.namecheap.com/support/knowledgebase/subcategory.aspx/46/domain-management>
* Flask - Web Development. (n.d.). Retrieved October 24, 2017, from <http://flask.pocoo.org/>
* Data.gov. (n.d.). Retrieved October 19, 2017, from <https://www.data.gov/>
* Front end development resources, Retrieved October 24, 2017, from <https://www.w3schools.com/html/>, <https://www.w3schools.com/css/> , and <https://www.w3schools.com/js/>
* Google maps resources, Retrieved 12/5/2017, from <https://developers.google.com/maps/documentation/javascript/tutorial>
* Chrome browser software,Retrieved 12/5/2017, from <https://www.google.com/chrome/browser/desktop/index.html>
* Firefox browser software,Retrieved 12/5/2017, from <https://www.mozilla.org/en-US/firefox/>
* Safari browser software,Retrieved 12/5/2017, from <https://support.apple.com/downloads/safari>

**2. System Overview**

The system overview for The Redistrictinator describes the overall system of the product.

**2.1 Background**

The Redistrictinator software is a tool that will allow users to view dynamically generated web content that represents possible districts that can be generated based on the users selection of state and number of districts. Operators and administrators of this software can use it to serve this functionality to any internet user that wishes to generate possible districts for any state in the US excluding Hawaii and Alaska.

Software maintainers, administrators, and operators can run this software on their own linux server, but this project was built using AWS infrastructure so it will cover how to run and operate this software using the same infrastructure. Since the software has not been load tested on a larger user base, occasionally there may be need to restart services, reset the system, and or recover from crashes. All of these functions can be done entirely through a remote shell environment that connects to the linux server.

**2.2 Hardware and Software Requirements**

This software was written to operate with as little software and hardware requirements as possible so that the largest number of users can utilize it.

The software was designed on and targeted to run on Ubuntu 16.04.3 LTS but should be operable on any computer with a linux operating system and an internet connection. The user interface was designed, targeted, and tested to work for Google Chrome(version 62.0.3202), Mozilla Firefox ( version 57.0), and Internet Explorer (version 11.0.1 ). The browser software is able to run on any operating system. Urls to acquire the browser software is located in Section 1.2 ( References)

This software was designed on and is targeted for servers with Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz and 1GB of RAM. Users can their browser software on any computer with a Core i3 CPU, and a minimum of 500 MB of RAM.

**3. Administrative Procedures**

The administrative procedures for The Redistrictinator details the process the system administrator should go through while maintaining the system.

**3.1 Installation**

Firstly, a T2 Micro instance on AWS can be acquired by following the directions at this url here: <https://aws.amazon.com/documentation/ec2/>. The documentation for AWS is very extensive and thorough on its own so it is not covered in this document. When configuring the instance, a minimum of 8 GB for the instance is required. All options can be left in their default configuration except for the Operating System (Should be Ubuntu), and the Security Groups, which should allow inbound/outbound traffic on ports 443, 80, and 22.

After configuring a server, set up and attach an elastic IP address to the instance by following these directions: <http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-eips.html>. The resulting IP address will be the public facing ip address for the server.

If you wish to, you can acquire a domain and point it to your public ip address by following the directions at this url: <https://www.namecheap.com/support/knowledgebase/subcategory.aspx/46/domain-management>

This software can be installed on the server by running the following commands after SSH-ing into the server.

|  |
| --- |
| git clone <https://github.com/sumanthneerumalla/BRICKS.git>  cd BRICKS/  sudo apt-get install python3.6  wget <https://bootstrap.pypa.io/get-pip.py>  sudo python3.6 get-pip.py  python3.6 -m pip install geopy flask  sudo nohup python3.6 server.py & |

The last command is the one that starts the python server backend and runs it in the background. This way the server will continue to run even after disconnecting your SSH session. All logs, error messages, and general output from the backend software will be stored in a file called nohup.out. This can be used to diagnose the software if any errors come up.

This handles the backend installation of the software. The frontend browser software can be installed using any of the following links:

* Chrome browser software: <https://www.google.com/chrome/browser/desktop/index.html>
* Firefox browser software : <https://www.mozilla.org/en-US/firefox/>
* Safari browser software: <https://support.apple.com/downloads/safari>

Browsers can now be pointed towards your backend servers ip address or domain name in order to start using the software.

**3.2 Routine Tasks**

Discuss any routine tasks that must be performed such as creating and maintaining user accounts.

The site should routinely be checked for performance and to ensure nothing is broken. The algorithm should similarly be checked regularly to ensure that it is returning and displaying the correct results.

Occasionally, the server may need to be restarted to perform operating system updates. When the server is restarted, the backend service can be restarted by sshing into the server again, going into the BRICKS software directory and then running the following command.

|  |
| --- |
| sudo nohup python3.6 server.py & |

Logs and metrics can be acquired by parsing through the file named nohup.out. Users and maintainers of this software can get a list of all ip addresses that visited the website, the time visited, type of request made, and urls visited by parsing this file. No support is provided by the authors of this software to parse this file. It is left to the end users to do what they wish to do with this file.

**3.3 Periodic Administration**

The main periodic task required for The Redistrictinator is updating zip code data as newer complete data is made available for use. Additionally the webapp should be tested and maintain functionality as newer updates to web browsers are released.

Log files (nohup.out) can be occasionally be backed up and/or deleted so that they don't take up all the hard drive space on the computer. In order to do this, the server must be stopped before making the backup of the log file. This can be done by running the following command after navigating to the BRICKS/ directory.

|  |
| --- |
| sudo killall python |

This kills the python process that started the server. It can be restarted after making a backup of the log file with the following command:

|  |
| --- |
| sudo nohup python3.6 server.py & |

Discuss any tasks to be performed periodically such as system backups and the cleaning up of user accounts.

**3.4 User Support**

There will be contact information information available to users on a dedicated page of the website. This will contain an email address with which support staff can be contacted in case of an issue.

The writers of this software provide absolutely no support other than this documentation that is provided. Since this project is open source however, users may open a github issue at this url: [https://github.com/sumanthneerumalla/BRICK](https://github.com/sumanthneerumalla/BRICKS) so that the open source community can address it.

**4. Troubleshooting**

The troubleshooting for The Redistrictinator describes the procedure to follow when correcting errors in the document.

**4.2 Dealing with Error Messages and Failures**

Occasionally, considering that the set of real world use scenarios are larger than the set of scenarios used to test this software, it is possible that serious error messages and failures may occur when running the software.

This software was designed to serve the homepage to multiple simultaneous users, but under high loads, it is possible for one user to receive the results of an other users map request. Under this circumstance, we recommend notifying users that the website is under high load, and that they should wait for a non-peak usage period.

If the python server crashes completely for some unforeseen reason, due to high load, or some out of the norm use case, it can simply be restarted with the following command:

|  |
| --- |
| sudo nohup python3.6 server.py & |

If this does not work, the entire project directory can be deleted and the entire software can be reset by following the directions in Section 3.1 - Installation.

**4.3 Known Bugs and Limitations**

Our software development process kept the core requirements of the software in mind and prioritized core features over vanity features, as a result, the number of bugs was kept to a minimum, and do not affect the vast majority of use cases. The bugs that do exist, are not fatal to the operation of the software. Below is a table of known bugs that were accounted for and explained during the presentation of the software

The first bug is that the generated districts are not always contiguous for certain states and district counts. This is located in file back.py on lines: 95-197. This bug exists because the districting generation algorithm needs additional refinement, and required more time to root out the edge cases where this could happen. This occurs only for certain states and when the user asks the system to generate a specific number of districts for that state. It can be dealt with by spending some more time on identifying the causes of the noncontiguous districts, and fixing the backend logic that creates them.

The second bug is that buttons on the webpage don't display with the right formatting for all browsers. This is located in all html/css files in lines 10-30. This bug exists because the organization of CSS styling has some mistakes in it, and they were too difficult to address in time for software delivery. We dealt with this by simply hiding the buttons from the users, so that the users experience of the software tool was not diminished.

The last bug is that user sessions can get mixed up with each other under high loads. It is located in file: server.py in lines 49-65. This bug exists because it was not foreseen until closer to the delivery date and individual user sessions were not a feature that was planned. It can later be addressed by adding cookies and individual user sessions so that the results of each individual district generation request can be directed only to the user that requested it.

**Appendix A – Team Review Sign-off**

All of the following team members have reviewed this document and agree to the content and format.

**Team Members:**

Benjamin Jeremenko: \_\_\_\_\_\_\_\_ Date: \_\_\_12/5/17\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jacob Philip: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_12/5/17\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sumanth Neerumalla: \_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_12/5/17\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Zachary Elliott: \_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_12/5/17\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Francis Kato: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_12/5/17\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Nathaniel Fuller: \_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_12/5/17\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Appendix B – Document Contributions**

Benjamin Jeremenko: Introduction 15%

Jacob Philip: Administrative Procedures 15%

Sumanth Neerumalla: System Overview, Administrative Procedures 30%

Zachary Elliott: System Overview 15%

Francis Kato: Troubleshooting, Appendices 10%

Nathaniel Fuller: Introduction, Troubleshooting 15%