

**Interactive Statistic Mapping Application**

**Software Test Report (STR)**

CMSC447\_Team3\_CodePods\_STR

Version 1.3

May 15, 2017

|  |  |
| --- | --- |
| **Course** | CMSC 447 |
| **Team** | Team 3 - Code Pods |
| **Members** | Kevin Miller  David Pan  Benjamin Hazlett  Desiree Mercuree  Darrell Laffoon  Ian Moskunas |
| **Sponsor (Customer)** | Shawn Squire |

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| 03/26/2018 | 1.0 | First Draft Started |
| 5/11/2018 | 1.1 | First Test Results completed |
| 5/14/2018 | 1.2 | Second Test Results completed |
| 5/15/2018 | 1.3 | Third Test Results completed |

**Table of Contents**

[**Introduction**](#_gjdgxs) **4**

[Purpose](#_94pw1ogunvju) 4

[Scope](#_at6dn5z70qz8) 4

[**References and Definitions**](#_gnz8qc44f5cj) **4**

[Table 1 - Document Artifacts](#_6dj1j7eavujw) 4

[Table 2 - Glossary of Terms](#_fjapkn1xv87d) 4

[**System Overview**](#_aj1pdb38zxvo) **6**

[Diagram 1 - System Overview](#_rbv2sdz15po) 6

[Table 3 - Application Components](#_nwsbzxy9zdi2) 6

[Table 4 - Application Actors](#_7g80tbsrr37n) 7

[**Overview of test results**](#_lnxbz9) **7**

[Overall assessment of the software tested](#_1ksv4uv) 7

[Impact of test environment](#_z337ya) 8

[Recommended improvements](#_3j2qqm3) 8

[**Detailed test results**](#_v2j7cbqp4776) **8**

[Test Results 05/15/2018](#_8mxoo8bydyw9) 8

[Test Results 05/14/2018](#_es6a2triomn) 10

[Test Results 05/11/2018](#_hg4wwjzijkl) 11

# Introduction

## Purpose

The purpose of this Software Test Report (STR) is to describe the results of testing the Statistics Mapping Application as defined in the Software Test Description (STD) document ((CMSC447\_Team3\_CodePods\_STD)

## Scope

This STR describes integration and regression tests, primarily from the user interface perspective. Unit test results are not included and are tracked by the individual developer or by the continuous integrations processes for any automated unit tests.

# References and Definitions

### *Table 1 - Document Artifacts*

|  |  |  |  |
| --- | --- | --- | --- |
| **Document** | **Description** | **Version** | **Date** |
| CMSC447\_Team3\_CodePods\_SRS | Software Requirements Specification | 1.2 | 03/26/2018 |
| CMSC447\_Team3\_CodePods\_SDP | Software Development Plan | 1.1 | 03/26/2014 |
| CMSC447\_Team3\_CodePods\_SDD | Software Design Description | 1.1 | 05/12/2018 |
| CMSC447\_Team3\_CodePods\_STD | Software Test Description | 1.1 | 05/14/2018 |
| CMSC447\_Team3\_CodePods\_STR | Software Test Report | 1.3 | 05/15/2018 |
| CMSC447\_Team3\_CodePods\_SUM | Software Users Manual | 1.2 | 03/26/2018 |

### 

### *Table 2 - Glossary of Terms*

|  |  |
| --- | --- |
| **Term** | **Meaning** |
| **The Application** | The targeted software solution - the Interactive Statistic Mapping Application |
| **The System** | The System that encompasses the application. The System and The Application could be used interchangeably in most cases |
| **The Customer** | The Customer that sponsored the project/software Application - Shawn Squire |
| **The Team** | The team of students who will build the Application |
| **SDLC** | Software Development Life Cycle - describes the steps and phases used to design, build and test the application |
| **Agile** | A set of principles that are used to define a iterative and incremental SDLC that is used to build the Application in iterations and allows for Customer feedback to guide the development of the Application |
| **Actor** | A person, external system, or other ‘actor’ who interacts with The Application or System |
| **Component** | A sub-system or part of the Application or System |
| **Map** | The selected, displayed Map, including boundaries. For example, it could be the State of Maryland, or Baltimore, or a neighborhood |
| **Map Data** | Detailed data about the selected Map |
| **Statistics Option** | One of the statistic choices such as crime, school ranking, etc. |
| **Proof of Concept** | A version of the Application that is used to test and prove aspects of the design |
| **Use Case** | Details behavioral requirements |
| **System Model** | Diagram that depicts system components and communication context |
| **Class Model** | Documents the Data entities of the System |
| **Sequence Model** | Documents the sequence of events between actors and components for particular use cases and application events |
| **State Model** | Documents state transitions for the system during particular uses casee and application events. |
| **Test Driven Development (TDD)** | A development methodology where unit tests are created before components are built. |
| **Software Repository** | A distributed data store that holds and tracks versions of the Application’s source code, documentation and other artifacts. |
| **Unit Test** | A test done on low level components as they are built |
| **Integration Test** | A test that tests the ability of system components to work together properly |
| **Regression Test** | A Test that checks that previously implemented features are still working after changes are deployed |

# 

# System Overview

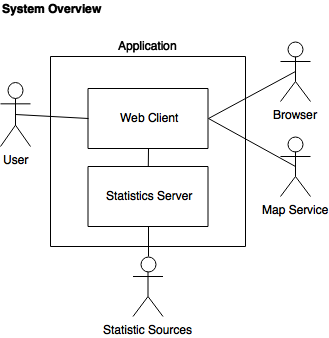
This section provides a high level System Overview for the Application. The Application’s purpose is to help a user answer a universal question:

"Where to I want to live, work, or retire?”

The Application will allow a user to select a Map and visually overlay important Statistics such as crime, income, school ranking, average commute, etc. The overlay will help the user identify areas on the Map that are more or less desirable. An expected, common use case would be a parent, who is looking to move to a safe and desirable place to raise her child, would use the Application to see and compare the crime rate and school rankings in the areas of interest.

*Diagram 1* depicts the key actors and components of the Application.

## *Diagram 1 - System Overview*

[](https://www.draw.io/#G1nqSkvUCma-uFmSX96F2KdXn3Pjgeq0vn)

The Application will consist of two components:

## *Table 3 - Application Components*

|  |  |
| --- | --- |
| **Component** | **Description** |
| **Web Client** | Frontend web application that allow a user to interactively identify a Map, and desired Statistics. It will dynamically update the Map with a visualization of the selected Map and selected Statistics Options |
| **Statistics Sources** | Backend server application that provides aggregated Statistical Data (to the Web Client), such as crime, income, commute, etc., from various trusted Statistic sources. This server encapsulates all data sources and converts them into a common, normalized format |

The Application will have of four primary actors :

## *Table 4 - Application Actors*

|  |  |
| --- | --- |
| **Actor** | **Description** |
| **User** | Will interact with the Web Client to select the Map and Statistics Option she wishes to see visualized |
| **Browser** | Will host the Application and provide input and output. Specifically, the browser will display the Map and a representation of the Statistical Data, etc. |
| **Map Service** | An external map service (Google Maps) that will provide the mapping capabilities and Map Data to the Web Client |
| **Statistic Services** | Publicly accessible Statistical Data sources that will return data for the user selected Map. *See Appendix for list of sources under consideration* |

The User will interact with the Web Client, which will present a Map and Statistics Options. As the User makes changes to the Map and/or chooses a Statistics Option, the Web Client will communicate with the Statistics Server to get updated Statistical Data for the selected Map. The Web Client will then redraw the Map to include Statistical Data based on the User’s selections. The Map Service will provide Map Data. The Statistics Sources will provide Statistics Data.

A detailed look at the user requirements of the Application and the corresponding functional and non-functional system requirements can be found in the Software Requirements Specification (SRS) document (CMSC447\_Team3\_CodePods\_SRS).

# Overview of test results

This section shall be divided into the following paragraphs to provide an overview of test results.

## Overall assessment of the software tested

As of 5/15/2018, the Application is presenting in production as a working minimal viable product. As required, the Application is demonstrating the following:

* The Application provides a simple, minimalistic and intuitive interface for the user
* The Web Client allows the user to select and manipulate the Map via interactive methods (mouse and touchscreen)
* The heatmap visualization of the Statistical Data is a gradient that ranges from Green (best value) to Red (worst value) for Crime and Commute
* There is a KML layer for Income that also varies from Red to Green
* Multiple external data sources are used
* The Statistics Options include working statistic data sources for Crime, Income and Commute
* During a User’s session, their selections persist at all times. Only the User can change the Map or the selected Statistic Option refreshed and displays an updated heatmap for the statistic

There are a few requirements that have not been met yet:

* Initially, School Ranking was a requirement. But an adequate data source was not procured in time, and using the discretion granted by the Customer, the team Pivoted to Income Data.
* The Customer also desired that the User could enter and Zip, State, etc, and select the map accordingly. This functionality does not yet exist.
* If the map area is too large, performance takes a hit, so further optimization is needed

## Impact of test environment

The test environment matches the production environment in terms of capabilities. So, there is little concern that there will be negative impact as the Application is deployed from test to production.

## Recommended improvements

The following are recommended improvements:

* Complete the Income Data Map, and also procure and implement School Ranking
* Color code the markers that represent the statistic data to correspond to the color coding based on their statistical ranking (red, green and yellow)
* Reflect an area the screen that displays the current statistic and the meaning of the color ranking. For example, show a legend that maps the Commute time to the color gradient.
* Add the search map by text box feature. Allowing zip, state, or other address to be used to navigate and redraw a new map

# Detailed test results

## 

## Test Results 05/15/2018

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Description | Pass/Fail | Comments |
| TD-01-001 | The User selects an area of the Map using the mouse | **Pass** | Able to use mouse to grab and move, zoom in/out and center on a point as expected |
| TD-01-002 | The User selects an area of the Map a search box | **Fail** | This Requirement has not yet been implemented |
| TD-02-001 | The User selects a Statistic from the Statistics Options list | **Pass** | USer able to select the statistic from the drop down and the map updates as expected to display the heatmap for the selected statistic |
| TD-02-002 | The User clears the Statistic from the Statistics Options list | **Pass** | The user can select an empty option in the metric selection dropdown to clear current metric data |
| TD-03-001 | Test that the proper map border is returned by the map service as changes to the map are made | **Pass** | Able to change the map and test that the border coordinates change accordingly using known borders as the test samples |
| TD-04-001 | Query the Crime API | **Pass** | Able to directly query the backend server with several known test samples and expected data is returned. Also able to use the Web Client to display the same ‘areas’ and am able to see the proper heatmap visualization |
| TD-04-002 | Query the Commute API | **Pass** | Commute data is now retrieved from the backend and displayed as a heatmap |
| TD-04-003 | Query the Income API | **Pass** | Income data is now fetched from the backend, and county boundaries are drawn and colored |
| TD-04-004 | Query the School Ranking API | **Fail** | School API not yet implemented and pushed to test environment |
| TD-05-001 | The user will select a new heatmap after a previous heatmap has been selected. | **Pass** | As map and statistics change the map properly redraws to reflect the selections made. |

## Test Results 05/15/2018

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Description | Pass/Fail | Comments |
| TD-01-001 | The User selects an area of the Map using the mouse | **Pass** | Able to use mouse to grab and move, zoom in/out and center on a point as expected |
| TD-01-002 | The User selects an area of the Map a search box | **Fail** | This Requirement has not yet been implemented |
| TD-02-001 | The User selects a Statistic from the Statistics Options list | **Pass** | USer able to select the statistic from the drop down and the map updates as expected to display the heatmap for the selected statistic |
| TD-02-002 | The User clears the Statistic from the Statistics Options list | **Pass** | There is no way to currently clear the statistic other than starting a new browser session |
| TD-03-001 | Test that the proper map border is returned by the map service as changes to the map are made | **Pass** | Able to change the map and test that the border coordinates change accordingly using known borders as the test samples |
| TD-04-001 | Query the Crime API | **Pass** | Able to directly query the backend server with several known test samples and expected data is returned. Also able to use the Web Client to display the same ‘areas’ and am able to see the proper heatmap visualization |
| TD-04-002 | Query the Commute API | **Pass** | Able to directly query the backend server with several known test samples and expected data is returned. Also able to use the Web Client to display the same ‘areas’ and am able to see the proper heatmap visualization |
| TD-04-003 | Query the Income API | **Pass** | Able to directly query the backend server with several known test samples and expected data is returned. Also able to use the Web Client to display the same ‘areas’ and am able to see the proper heatmap visualization |
| TD-04-004 | Query the School Ranking API | **Fail** | School Ranking API not yet implemented and pushed to test environment |
| TD-05-001 | The user will select a new heatmap after a previous heatmap has been selected. | **Pass** | As map and statistics change the map properly redraws to reflect the selections made. |

## Test Results 05/14/2018

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Description | Pass/Fail | Comments |
| TD-01-001 | The User selects an area of the Map using the mouse | **Pass** | Able to use mouse to grab and move, zoom in/out and center on a point as expected |
| TD-01-002 | The User selects an area of the Map a search box | **Fail** | This Requirement has not yet been implemented |
| TD-02-001 | The User selects a Statistic from the Statistics Options list | **Pass** | User able to select the statistic from the drop down and the map updates as expected to display the heatmap for the selected statistic |
| TD-02-002 | The User clears the Statistic from the Statistics Options list | **Fail** | There is no way to currently clear the statistic other than starting a new browser session |
| TD-03-001 | Test that the proper map border is returned by the map service as changes to the map are made | **Pass** | Able to change the map and test that the border coordinates change accordingly using known borders as the test samples |
| TD-04-001 | Query the Crime API | **Pass** | Able to directly query the backend server with several known test samples and expected data is returned. Also able to use the Web Client to display the same ‘areas’ and am able to see the proper heatmap visualization |
| TD-04-002 | Query the Commute API | **Pass** | Able to directly query the backend server with several known test samples and expected data is returned. Also able to use the Web Client to display the same ‘areas’ and am able to see the proper heatmap visualization |
| TD-04-003 | Query the Income API | **Fail** | Income API not yet implemented and pushed to test environment |
| TD-04-004 | Query the School Ranking API | **Fail** | School Ranking API not yet implemented and pushed to test environment |
| TD-05-001 | The user will select a new heatmap after a previous heatmap has been selected. | **Pass** | As map and statistics change the map properly redraws to reflect the selections made. |

## Test Results 05/11/2018

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Description | Pass/Fail | Comments |
| TD-01-001 | The User selects an area of the Map using the mouse | **Pass** | Able to use mouse to grab and move, zoom in/out and center on a point as expected |
| TD-01-002 | The User selects an area of the Map a search box | **Fail** | This Requirement has not yet been implemented |
| TD-02-001 | The User selects a Statistic from the Statistics Options list | **Pass** | USer able to select the statistic from the drop down and the map updates as expected to display the heatmap for the selected statistic |
| TD-02-002 | The User clears the Statistic from the Statistics Options list | **Fail** | There is no way to currently clear the statistic other than starting a new browser session |
| TD-03-001 | Test that the proper map border is returned by the map service as changes to the map are made | **Pass** | Able to change the map and test that the border coordinates change accordingly using known borders as the test samples |
| TD-04-001 | Query the Crime API | **Pass** | Able to directly query the backend server with several known test samples and expected data is returned. Also able to use the Web Client to display the same ‘areas’ and am able to see the proper heatmap visualization |
| TD-04-002 | Query the Commute API | **Fail** | Commute Map is still showing test map, not updating dynamically |
| TD-04-003 | Query the Income API | **Fail** | Income API not yet implemented and pushed to test environment |
| TD-04-004 | Query the School Ranking API | **Fail** | School Ranking API not yet implemented and pushed to test environment |
| TD-05-001 | The user will select a new heatmap after a previous heatmap has been selected. | **Pass** | As map and statistics change the map properly redraws to reflect the selections made. |

## 