

**Interactive Statistic Mapping Application**

**Software Test Description (STD)**

CMSC447\_Team3\_CodePods\_STD

Version 1.1

May 14, 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 |
| 05/14/2018 | 1.1 | First version completed |

**Table of Contents**

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

[Purpose](#_5wkzo6sbs1qg) 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) 7

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

[**Test preparations**](#_o4b8k2ceb2l) **7**

[Unit Testing](#_zg3ml6gmyb4o) 8

[Integration Testing](#_gbawjap2a3h7) 8

[Regression Testing](#_rc9qc9wedz9q) 8

[**Test Descriptions**](#_xty53tm9j42e) **8**

[TD-01 Select Map Tests](#_2f8zwa24c2x6) 9

[TD-02 Select Statistics Option Tests](#_vf6795eycj77) 10

[TD-03 Provide Map Data Tests](#_u12odqrxnufy) 11

[TD-04 Provide Statistical Data Tests](#_g6pkwg1iz5gt) 11

[TD-05 Redraw Map Tests](#_29afiujhddl1) 12

# 

# Introduction

## Purpose

The purpose of this Software Test Description (STD) is to describe the test plan for the Interactive Statistic Mapping Application.

## Scope

This STD describes the detailed plan, testing methodology, and lists the individual tests that will be used to verify the Application is functioning as designed and is meeting known requirements.

The STD contains the following information:

* A systems overview
* Testing process description
* List of UX/Integration tests to be performed

# 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 | 03/26/2018 |
| 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 | 05/14/2018 |

### 

### *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

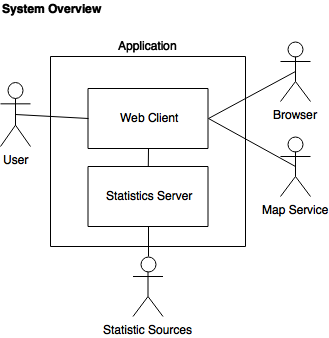
This section provides a high level System Overview for the Application. The Application’s purpose is to help a user answer a universal question:

"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.

# 

# Test preparations

There will be three phases of testing used during the Application development process:

* **Unit** - individual, low-level, code subunit are tested individually, during development.
* **Integration** - system components are tested to ensure they integrate properly.
* **Regression** - system level testing to ensure all requirements are met and previously completed functionality is still working as expected.

The next sections define the process used for each of the three phases.

## Unit Testing

The unit testing frameworks used are Mocha and Chai. All tests will go in the src/tests directory. Client unit tests will test the following:

* Ensure that the correct html is rendered dependent on the properties passed to the component.

Server unit tests will test the following:

* All endpoints return correct JSON objects appropriate for the endpoint
* Ensure that all external API requests function as expected

Any bugs found must have a corresponding unit test to prevent future regressions

## Integration Testing

The project contains a pipeline consisting of servers for Development, Staging, and Production. During development, the Development server will be synced with the working codebase at all times. When the team believes a release candidate is ready, the code will be merged to the master branch which will automatically be deployed to the Staging server provided all unit tests pass. The team will then test every implemented feature in the requirements document. Load tests will also be performed to ensure the expected number of users can operate the site concurrently. If the staging environment passes all tests and is approved by the team, then the code will be merged to the Production branch to deploy the the Production server.

## Regression Testing

The continuous integration tool, TravisCI, will automatically run the project unit tests whenever a commit is made to a main branch, development, master, or production. If a test fails, the team will be notified via Slack to prompt them to resolve the issue.

# Test Descriptions

The tests in this document are based on the user requirements of the Application, and the corresponding functional and non-functional system requirements, which can be found in the Software Requirements Specification (SRS) document (CMSC447\_Team3\_CodePods\_SRS). Tests are also derived to test aspects of the architecture and design as documented in the Software Design Description (SDD) document (CMSC447\_Team3\_CodePods\_SDD).

For all Tests described, a general assumption is that the Web Client and Statistics Server are running in good state and are accessible to the User’s web browser. Also, the User has started entered the URL of the Web Client and the web browser has rendered the Application, starting in it’s default state as described in the SRS.

## TD-01 Select Map Tests

### 

|  |  |  |
| --- | --- | --- |
| **Test ID** | TD-01-001 |  |
| Test description | The User selects an area of the Map using the mouse |  |
| Use Case ID | UC-01 |  |
| **Test Procedure** |  |  |
| **Step** | **Operator Action** | **Expected Results** |
| 1 | Drag the Map:  Point to a section of the displayed map and hold the left mouse button. While holding the right button move the mouse around | The Map will be redrawn such that the point of the map being ‘held’, moves with the mouse pointer allowing the user to scroll the map in any direction  The scrolling action stops when the left mouse button is released  Corresponding Statistics visualizations & the Key/Legend will also be updated |
| 2 | Zoom in to a point on the Map:  Point to a point on the map and double click the left mouse button | The map will recenter the map so the selection point is in the middle of the Map display area and the map will zoom in  Corresponding Statistics visualizations & the Key/Legend will also be updated  Assumptions: The zoom percentage will be what the Map Service provides by default, and the Map Service will also control how far the User can zoom in |
| 3 | Zoom out based on a point of the Map:  Point to a point on the map and double click the right mouse button | The map will recenter the map so the selection point is in the middle of the Map display area and the map will zoom out  Corresponding Statistics visualizations & the Key/Legend will also be updated  Assumptions: The zoom percentage will be what the Map Service provides by default, and the Map Service will also control how far the User can zoom out |

### 

|  |  |  |
| --- | --- | --- |
| **Test ID** | TD-01-002 |  |
| Test description | The User selects an area of the Map a search box |  |
| Use Case ID | UC-01 |  |
| **Test Procedure** |  |  |
| **Step** | **Operator Action** | **Expected Results** |
| 1 | The user navigates to the Map Search Box and enters a valid Zip Code, or City/State combination and hits enter or clicks on the search icon | The Map will be redrawn to reflect the search criteria entered  Assumption: The Map Service will evaluate the search criteria and will return it’s interpretation of the search. |
| 2 | The user navigates to the Map Search Box and enters an valid Zip Code, or City/State combination and hits enter or clicks on the search icon | The UI displays a message that there was no match found and the Map display is not update - the current selected Map and correstpoing UI components remain unchanged. |

## 

## TD-02 Select Statistics Option Tests

### 

|  |  |  |
| --- | --- | --- |
| **Test ID** | TD-02-001 |  |
| Test description | The User selects a Statistic from the Statistics Options list |  |
| Use Case ID | UC-02 |  |
| **Test Procedure** |  |  |
| **Step** | **Operator Action** | **Expected Results** |
| 1 | The user selects a Statistics Option from the list | The displayed Map reflects the selected Map and a visualization of the selected Statistics Option and an updated Key/Legend will be displayed to reflect the new Statistics state |

### 

|  |  |  |
| --- | --- | --- |
| **Test ID** | TD-02-002 |  |
| Test description | The User clears the Statistic from the Statistics Options list |  |
| Use Case ID | UC-02 |  |
| **Test Procedure** |  |  |
| **Step** | **Operator Action** | **Expected Results** |
|  |  |  |
| 1 | The user clears the Statistics Option | The displayed Map reflects the selected Map and Statistical visualization will be removed, and an updated Key/Legend will be displayed to reflect the new Statistics state |

## 

## TD-03 Provide Map Data Tests

Tests that will prove the Map Service is returning the proper data related to the selected Map.

### 

|  |  |  |
| --- | --- | --- |
| **Test ID** | TD-03-001 |  |
| Test description | Test that the proper map border is returned by the map service as changes to the map are made |  |
| Use Case ID | UC-03 |  |
| **Test Procedure** |  |  |
| **Step** | **Operator Action** | **Expected Results** |
|  |  |  |
| 1 | The user does any of the following,   * Zoom the map * Move the map * or cause the map to change via a search box (if presented) | The map will render and the border coordinates are returned for the updated map.  Using the Dev Tools, the border selected will be made available. Visual inspection will be made to ensure that the border coordinates match expected values as compared to a list of known borders.  Primarily we want to make sure the border values change as the map changes. The actually values returned come from Google API and are highly trustworthy. |

## 

## TD-04 Provide Statistical Data Tests

Tests that will prove the statistics coming back from the statistics server are for the right map area

### 

|  |  |  |
| --- | --- | --- |
| **Test ID** | TD-04-001 |  |
| Test description | Query the Crime API |  |
| Use Case ID | UC-04 |  |
| **Test Procedure** |  |  |
| **Step** | **Operator Action** | **Expected Results** |
|  |  |  |
| 1 | Provide several query strings to the Crime API | Test the results against precalculated responses |
| 2 | Use the client to select the same test map areas with the Crime Statistic Selected | The returned map includes the expected statistical data points |

## 

### 

|  |  |  |
| --- | --- | --- |
| **Test ID** | TD-04-002 |  |
| Test description | Query the Commute API |  |
| Use Case ID | UC-04 |  |
| **Test Procedure** |  |  |
| **Step** | **Operator Action** | **Expected Results** |
|  |  |  |
| 1 | Provide several query strings to the Commute API | Test the results against precalculated responses |
| 2 | Use the client to select the same test map areas with the Commute Statistic Selected | The returned map includes the expected statistical data points |

## 

### 

|  |  |  |
| --- | --- | --- |
| **Test ID** | TD-04-003 |  |
| Test description | Query the Income API |  |
| Use Case ID | UC-04 |  |
| **Test Procedure** |  |  |
| **Step** | **Operator Action** | **Expected Results** |
|  |  |  |
| 1 | Provide several query strings to the Income API | Test the results against precalculated responses |
| 2 | Use the client to select the same test map areas with the Income Statistic Selected | The returned map includes the expected statistical data points |

### 

|  |  |  |
| --- | --- | --- |
| **Test ID** | TD-04-004 |  |
| Test description | Query the School Ranking API |  |
| Use Case ID | UC-04 |  |
| **Test Procedure** |  |  |
| **Step** | **Operator Action** | **Expected Results** |
|  |  |  |
| 1 | Provide several query strings to the School Ranking API | Test the results against precalculated responses |
| 2 | Use the client to select the same test map areas with the Income Statistic Selected | The returned map includes the expected statistical data points |

## 

## TD-05 Redraw Map Tests

Tests that will prove that the Redrawn Map shows the current selected Map, the current selected Statistic, a heat map that reflects the Statistics Data and the the proper Key/Legend

### 

|  |  |  |
| --- | --- | --- |
| **Test ID** | TD-05-001 |  |
| Test description | The user will select a new heatmap after a previous heatmap has been selected. |  |
| Use Case ID | UC-05 |  |
| **Test Procedure** |  |  |
| **Step** | **Operator Action** | **Expected Results** |
| 1 | Select first heatmap:  Click on a heatmap option from the dropdown menu | The Map will display the correct statistical data stored. There will also be an update for the key/legend about what statistic is selected. |
| 2 | Select second heatmap:  Click on a new heatmap option from the dropdown menu | The Map will no longer display the old data and instead update to the correct data selected. The key/legend will also reflect this update. |

### 