CMSC 447

Software Test Description (STD)

[**Scope**](#_3znysh7) **4**

[Identification](#_1t3h5sf) 4

[System overview](#_4d34og8) 4

[Document overview](#_2s8eyo1) 4

[**Referenced documents**](#_3rdcrjn) **4**

[**Test preparations**](#_lnxbz9) **4**

[T0 - (Front End)](#_mw3gghwgytwz) 4

[Software preparation](#_hhw24uf42mno) 4

[Other pre-test preparations -- If there aren’t any remove](#_8v6p0qphfr4s) 5

[T1 - (Back End)](#_1ksv4uv) 5

[Software preparation](#_3j2qqm3) 5

[Other pre-test preparations -- If there are none then remove this section](#_2xcytpi) 5

[**Test descriptions**](#_3whwml4) **5**

[T2 - (Front End)](#_ufzhsh8k67t6) 5

[T2.1 - (Correct Query)](#_roadumrjnjrf) 5

[Requirements addressed -- (This covers ALL of our parameter requirements)](#_sguo15lujwj0) 6

[Prerequisite conditions](#_g7lluthi8qq4) 6

[Test inputs](#_exe5r4dqoojy) 6

[Expected test results](#_pmpa46jgrrot) 6

[Criteria for evaluating results](#_m7zo5zf3qqhw) 6

[Test procedure](#_iv2sepxmzgnd) 7

[Assumptions and constraints](#_hav5slpyaz76) 8

[T2.2 - (Working GUI)](#_xctmjxbjxbu0) 8

[Requirements addressed -- (This should cover all of our UI requirements)](#_9i5pk787m2j) 8

[Prerequisite conditions](#_91coo8ikhtaw) 8

[Test inputs](#_btql0gjk6cub) 8

[Expected test results](#_itu1rru69hjk) 9

[Criteria for evaluating results](#_uybhz6j37pdk) 9

[Test procedure](#_y79wnzsj3r9m) 9

[Assumptions and constraints](#_t63v11ple9iy) 10

[T3 - (Back End)](#_qsh70q) 10

[T3.1 - (Correct Results)](#_1pxezwc) 10

[Requirements addressed -- (Although this is more design than requirement, it should handle any correctness components)](#_2e4otj5ifs8o) 10

[Prerequisite conditions](#_g5s1apub1iu0) 11

[Test inputs](#_y0gll6cqq9ye) 11

[Expected test results](#_316kdeezvb94) 11

[Criteria for evaluating results](#_eooqacb45fx6) 11

[Test procedure](#_b311safnhlht) 12

[Assumptions and constraints](#_1qvd24zb97js) 12

[T3.2 - (Speed)](#_pi5xaxatjbhr) 13

[Requirements addressed -- (Covers speed requirements)](#_gsyvenf0uwul) 13

[Prerequisite conditions](#_vnx99sn4ttdd) 13

[Test inputs](#_koqt5paq6oj2) 13

[Expected test results](#_xpcof1l5axr) 14

[Criteria for evaluating results](#_6iiif3xu4d4o) 14

[Test procedure](#_vki3aybm9e9) 14

[Assumptions and constraints](#_h9otxem7qa5k) 15

[**Requirements traceability**](#_41mghml) **15**

[**Notes**](#_2grqrue) **15**

[**Appendixes**](#_vx1227) **17**

# Scope

## System overview

The system aims to provide information about areas to move to within the United States consistent with specified criteria. Each specified criteria will appear as a search option on a webpage and will be used by a user to narrow down results for a new location to move to. Locations fitting the criteria of the search will be displayed on a map as points.

## Document overview

This document outlines testing procedures for all system components. Each component is associated with a unique project identifier and given a brief description along with a comprehensive procedure for evaluating that component.

# Referenced documents

Documents referred to are the Software Design Description (SDD) and the Software Requirements Specification (SRS); when referred to their acronym and section number will be used to identify relevant information. All code will be stored on Github, with instructions on installation included.

# Test preparations

The following are preparations taken going forward into testing. Safety precautions if they exist will be marked with CAUTION. As there is no personal information being put on or saved by the project, privacy concerns are minimal.

## T0 - (Front End)

Front-end requirements include CSCI components I1-I19 as listed in section 4.1 of the SDD. This includes the user interface of the website and the displayed map. Preparations are the completion of the Front End, and installation of the Safari Web Browser to display the Front end.

### Software preparation

Preparation involves writing the HTML file and CSS file for the user interface as well as a JS file which will use the Google Maps API (A2) to display the search results as pins on the map.

The installation of the Front End website and the Back End local server are available on Github in a public repository, with instructions on how to set up the project on a local machine.

## T1 - (Back End)

Setup of the Back End is detailed on the Github, and is as follows. A local server must be established on the machine, and will be populated with the database described above. To run the local server, the user must install Apache, SQL, and PHP. For a Windows machine, wampserver will be used and the necessary repository from Github moved onto the server. From there, a local copy of the webpage can be run. on Unix machines, a similar method is detailed.

### Software preparation

The database will be built by querying the datausa.io API and collecting the information, followed by collecting weather and outdoor recreation data from NOAA weather stations and the RIDB API as CSV files. These files will then be integrated into a local SQL database for faster searching.

A file named search.php will recieve search criteria from the Front End and query the database for matching locations, and return the relevant results to the Front End for display.

# Test descriptions

This section shall be divided into the following paragraphs. Safety precautions, marked by WARNING or CAUTION, and security and privacy considerations shall be included as applicable.

## T2 - (Front End)

This paragraph shall identify a test by project-unique identifier and shall be divided into the following subparagraphs. When the required information duplicates information previously provided, that information may be referenced rather than repeated.

### T2.1 - (Correct Query)

This paragraph shall identify a test case by project-unique identifier, state its purpose, and provide a brief description. The following subparagraphs shall provide a detailed description of the test case.

The purpose of this test is to verify that for each parameter in the search menu, the correct information is sent to the database

#### Requirements addressed -- (This covers ALL of our parameter requirements)

The system requirements addressed by this test case are referenced in section 2.1.1 of the SRS. All requirements specified in the SRS shall be present and functional in the project.

#### Prerequisite conditions

There are no specific hardware or software requirements beyond the ability to run the Safari Web browser. For each parameter listed in 2.1.1 of the SRS, a full test will take place in which parameters are set to every combination possible

#### Test inputs

This paragraph shall describe the test inputs necessary for the test case. The following shall be provided, as applicable:

1. Name, purpose, and description (e.g., range of values, accuracy) of each test input

|  |  |  |
| --- | --- | --- |
| Name | Input subdivision | Description |
| Public School | High School | Indicates need for high graduation rates above <value> |
| Public Transport | Low Availability | Indicates need for low availability of public transport between 0<unit> and <low value> <unit> |
| Average Availability | Indicates need for average availability of public transport between <low value><unit> and <average value><unit> |
| High Availability | Indicates need for high availability of public transport between <average value><unit> and <high value><unit> |
| Crime | Low Crime Rate | Indicates need for low crime rate below <low value> number of violent crimes |
| Average Crime Rate | Indicates need for average crime rate between <low value> and <high value> number of violent crimes |
| Outdoor Recreation | Hiking | Indicates need for hiking areas within county |
| Climbing | Indicates need for climbing areas within county |
| Camping | Indicates need for camping areas within county |
| Fishing | Indicates need for fishing areas within county |
| Wilderness | Indicates need for wilderness areas within county |
| Climate | No Preference | Indicates that user has no preference for either”Hotter” or “Colder” Environment |
| Hotter | Indicates that user want annual temperatures above or equal to 54.45 degrees fahrenheit |
| Colder | Indicates that user wants annual temperatures below 54.45 degrees fahrenheit |
| Low Annual Precipitation | Indicates that user wants annual precipitation below <value> inches |
| Average Annual Precipitation | Indicates that user wants annual precipitation between <low value> and <average value> inches |
| High Annual Precipitation | Indicates that user wants annual precipitation above <average value> |
| Low Annual Snowfall | Indicates that user wants annual snowfall to be below <low value> inches |
| Average Annual Snowfall | Indicates that user wants annual snowfall to be between <low value> and <average value> inches |
| High Annual Snowfall | Indicates that user wants annual snowfall to be above <average value> inches |
| Health Care | Average number of doctors to patients | Indicates that user requests the ratio of doctors to patients be between <low value> and <average value> |
| High number of doctors to patients | Indicates that user requests the ratio of doctors to patients be above <average value> |
| Commute Time | --- | User indicates a value between 0 to 60 minutes for average commute time |

1. Source of the test input and the method to be used for selecting the test input  
   Test input will be selected by selecting all combinations of queries that the above list permits
2. Real or simulated test input:  
   Test input will be simulated via script
3. Event sequence of test input:
   1. Event 1: Choose permutation
   2. Event 2: Select input
   3. Event 3: Send to data
   4. Event 4: Confirm that permutation sent to database matches selected input
   5. Event 5: Repeat for all permutations
4. The manner in which the input data will be controlled to:

Test the item(s) with a minimum/reasonable number of data types and values

Exercise the item(s) with a range of valid data types and values that test for overload, saturation, and other "worst case" effects

Exercise the item(s) with invalid data types and values to test for appropriate handling of irregular inputs

Permit retesting, if necessary

#### Expected test results

The expected test result is that each of the inputs appears in the database as the user requested. E.g. If a user were to input that they wanted low annual snowfall and average crime rates the only queries sent to the database would appear asking for counties with low annual snowfall and average crime rates.

#### Criteria for evaluating results

This paragraph shall identify the criteria to be used for evaluating the intermediate and final results of the test case. For each test result, the following information shall be provided, as applicable:

* + - * 1. The range or accuracy over which an output can vary and still be acceptable:  
           As long as the input sent the to database is within the range defined in table above it is acceptable
        2. Minimum number of combinations or alternatives of input and output conditions that constitute an acceptable test result:  
           Testing of all permutations an input can go through constitutes as an acceptable test result
        3. Maximum/minimum allowable test duration, in terms of time or number of events:  
           Minimum allowable test duration is for each input
        4. Maximum number of interrupts, halts, or other system breaks that may occur:  
           Halts may only occur between inputs
        5. Allowable severity of processing errors:  
           Minimal severity of processing errors is acceptable; the site shall encounter no errors that would cause a failure to display data
        6. Conditions under which the result is inconclusive and re-testing is to be performed:  
           If a test halts halfway through an input, or no feedback is shown on the screen corrections will be performed and retested
        7. Conditions under which the outputs are to be interpreted as indicating irregularities in input test data, in the test database/data files, or in test procedures:  
           If the input does not match the query present in the database then output is assumed to have irregularity
        8. Allowable indications of the control, status, and results of the test and the readiness for the next test case (may be output of auxiliary test software)  
           If the test for the input returns without stopping in between them:
        9. Additional criteria not mentioned above.

#### Test procedure

This paragraph shall define the test procedure for the test case. The test procedure shall be defined as a series of individually numbered steps listed sequentially in the order in which the steps are to be performed. For convenience in document maintenance, the test procedures may be included as an appendix and referenced in this paragraph. The appropriate level of detail in each test procedure depends on the type of software being tested. For some software, each keystroke may be a separate test procedure step; for most software, each step may include a logically related series of keystrokes or other actions. The appropriate level of detail is the level at which it is useful to specify expected results and compare them to actual results. The following shall be provided for each test procedure, as applicable:

1. Test operator actions and equipment operation required for each step, including commands, as applicable, to:
2. Initiate the test case and apply test inputs
3. Inspect test conditions
4. Perform interim evaluations of test results
5. Record data
6. Halt or interrupt the test case
7. Request data dumps or other aids, if needed
8. Modify the database/data files
9. Repeat the test case if unsuccessful
10. Apply alternate modes as required by the test case
11. Terminate the test case
12. Expected result and evaluation criteria for each step
13. If the test case addresses multiple requirements, identification of which test procedure step(s) address which requirements. (Alternatively, this information may be provided in [5.)](#_41mghml)
14. Actions to follow in the event of a program stop or indicated error, such as:
15. Recording of critical data from indicators for reference purposes
16. Halting or pausing time-sensitive test-support software and test apparatus
17. Collection of system and operator records of test results

1. Procedures to be used to reduce and analyze test results to accomplish the following, as applicable:

Detect whether an output has been produced

Identify media and location of data produced by the test case

Evaluate output as a basis for continuation of test sequence

Evaluate test output against required output

#### Assumptions and constraints

Below are identified any assumptions made and constraints or limitations imposed in the description of the test case due to system or test conditions, such as limitations on timing, interfaces, equipment, personnel, and database/data files. If waivers or exceptions to specified limits and parameters are approved, they shall be identified and this paragraph shall address their effects and impacts upon the test case.

### T2.2 - (Working GUI)

The purpose of this test is to verify that each of the interface objects, labelled (I#) in the SDD, works as intended. The following subparagraphs will include a description of the test case.

#### Requirements addressed -- (This should cover all of our UI requirements)

The requirements that will be tested are defined in the Software Design Document (SDD); alternatively they may be referenced in section 5.a.

#### Prerequisite conditions

No prerequisite conditions exist for the project, as the API’s are from trusted sources such as Google and the United States Government.

#### Test inputs

Test inputs are listed below based on the requirements provided in the SDD sections 4.1 CSCI Components; 4.3 Interface Design; and 5.1 Value range descriptions and database sources.

* All components defined in SDD 4.1 shall be present in the project front end
* Interfaces as defined in SDD 4.3 will be present and functional as defined
* Value range descriptions shall be as stated in SDD 5.1, with clarifications from feedback below:

1. (I15, I16) Weather data shall be a check box system for temperature, climate, and weather for presets; changed from rainfall and temperature sliders.
2. (I19) Commute Time slider adjusted from [0,100] range to [0,60] range to reflect an average commute time, from 0 minutes to one hour (60 minutes).

* Text pop ups will appear next to parameters, so as to define their purpose to the user when hovered over.
* If data is not available from the database, the location will still be added to the map. However, a maker will be added that when hovered over explains that some data is missing to provide feedback to the user.
* All variables sent to the database will be with in the acceptable range defined by the sliders and checkboxes, as defined in SDD 5.1.

#### Expected test results

The test results are expected to reflect the requirements set forth in the SDD and above. Final test results shall reflect the changes mentioned above, and any further clarifications set forward by the customer, which will be annotated as they occur.

#### Criteria for evaluating results

Below are the intermediate and final test criteria. Intermediate tests will serve as checkpoints as the project is developed to ensure that the site is displaying properly. The final test will consist of repeated runs of the project from a minimum of five random users to ensure that accurate and diverse results are occurring, and are properly represented on the map.

* As above, ensure parameters are within the tolerable limits
* All possible combinations of parameters are checked
* The results will be returned in the acceptable time limit defined in the SDD, not to exceed the maximum of fifteen (15) seconds
* Any errors are captured and handled, and an error message displayed
* The site will remain stable in the event of an error; the parameters in question will be reset to their defaults and resubmission allowed
* Any errors encountered will be logged in the Software Test Report (SRT) and corrected
* The site will be stable and fit the above criteria, in addition to working on the Safari web browser

#### Test procedure

The intermediate tests will be performed by the developers as they are implemented to ensure basic functionality. The final test will be run by the team members on their individual machines in the Safari web browser on all parameters and compared to ensure that consistent results are achieved. Any discrepancies will be recorded in the SRS and corrected. Following this initial test to confirm the site is stable and accurate, a battery of at least 5 random tests will be done by presenting the site to random individuals known to the developers. The purpose of the site will be explained and the individuals allowed to use the site. This is to ensure that the site continues to be stable and is user friendly to others not involved in the development process.

#### Assumptions and constraints

A series of constraints are related to the testing, namely in that the project is hosting locally. All necessary files are on the public github and available for all to download and install; provided that the instructions are followed errors should be isolated to the code developed and not missing resources. In addition while the project will be able to run in other web browsers, at a minimum it shall run in the Safari web browser. it is assumed that the user will have all necessary resources downloaded from the github and the Safari web browser installed to ensure the project operates properly.

## T3 - (Back End)

This paragraph shall identify a test by project-unique identifier and shall be divided into the following subparagraphs. When the required information duplicates information previously provided, that information may be referenced rather than repeated.

### T3.1 - (Correct Results)

The purpose of this test is to verify that the results (counties) returned by the back end, using information from the database, coincide with the known results of a few counties. This will allow verification of database correctness and integrity.

#### Requirements addressed -- (Although this is more design than requirement, it should handle any correctness components)

This paragraph shall identify the CSCI or system requirements addressed by the test case. (Alternatively, this information may be provided in [5.a.)](#_41mghml)

#### Prerequisite conditions

This paragraph shall identify any prerequisite conditions that must be established prior to performing the test case. The following considerations shall be discussed, as applicable:

* + - * 1. Hardware and software configuration
        2. Flags, initial breakpoints, pointers, control parameters, or initial data to be set/reset prior to test commencement
        3. Preset hardware conditions or electrical states necessary to run the test case
        4. Initial conditions to be used in making timing measurements
        5. Conditioning of the simulated environment
        6. Other special conditions peculiar to the test case

#### Test inputs

This paragraph shall describe the test inputs necessary for the test case. The following shall be provided, as applicable:

1. Name, purpose, and description (e.g., range of values, accuracy) of each test input
2. Source of the test input and the method to be used for selecting the test input
3. Whether the test input is real or simulated
4. Time or event sequence of test input
5. The manner in which the input data will be controlled to:

Test the item(s) with a minimum/reasonable number of data types and values

Exercise the item(s) with a range of valid data types and values that test for overload, saturation, and other "worst case" effects

Exercise the item(s) with invalid data types and values to test for appropriate handling of irregular inputs

Permit retesting, if necessary

#### Expected test results

Below are identified all expected test results for the test case. Both intermediate and final test results shall be provided, as applicable.

#### Criteria for evaluating results

Below are identified the criteria to be used for evaluating the intermediate and final results of the test case. For each test result, the following information shall be provided, as applicable:

* + - * 1. The range or accuracy over which an output can vary and still be acceptable
        2. Minimum number of combinations or alternatives of input and output conditions that constitute an acceptable test result
        3. Maximum/minimum allowable test duration, in terms of time or number of events
        4. Maximum number of interrupts, halts, or other system breaks that may occur
        5. Allowable severity of processing errors
        6. Conditions under which the result is inconclusive and re-testing is to be performed
        7. Conditions under which the outputs are to be interpreted as indicating irregularities in input test data, in the test database/data files, or in test procedures
        8. Allowable indications of the control, status, and results of the test and the readiness for the next test case (may be output of auxiliary test software)
        9. Additional criteria not mentioned above.

#### Test procedure

This paragraph shall define the test procedure for the test case. The test procedure shall be defined as a series of individually numbered steps listed sequentially in the order in which the steps are to be performed. For convenience in document maintenance, the test procedures may be included as an appendix and referenced in this paragraph. The appropriate level of detail in each test procedure depends on the type of software being tested. For some software, each keystroke may be a separate test procedure step; for most software, each step may include a logically related series of keystrokes or other actions. The appropriate level of detail is the level at which it is useful to specify expected results and compare them to actual results. The following shall be provided for each test procedure, as applicable:

1. Test operator actions and equipment operation required for each step, including commands, as applicable, to:
2. Initiate the test case and apply test inputs
3. Inspect test conditions
4. Perform interim evaluations of test results
5. Record data
6. Halt or interrupt the test case
7. Request data dumps or other aids, if needed
8. Modify the database/data files
9. Repeat the test case if unsuccessful
10. Apply alternate modes as required by the test case
11. Terminate the test case
12. Expected result and evaluation criteria for each step
13. If the test case addresses multiple requirements, identification of which test procedure step(s) address which requirements. (Alternatively, this information may be provided in [5.)](#_41mghml)
14. Actions to follow in the event of a program stop or indicated error, such as:
15. Recording of critical data from indicators for reference purposes
16. Halting or pausing time-sensitive test-support software and test apparatus
17. Collection of system and operator records of test results
18. Procedures to be used to reduce and analyze test results to accomplish the following, as applicable:

Detect whether an output has been produced

Identify media and location of data produced by the test case

Evaluate output as a basis for continuation of test sequence

Evaluate test output against required output

#### Assumptions and constraints

This paragraph shall identify any assumptions made and constraints or limitations imposed in the description of the test case due to system or test conditions, such as limitations on timing, interfaces, equipment, personnel, and database/data files. If waivers or exceptions to specified limits and parameters are approved, they shall be identified and this paragraph shall address their effects and impacts upon the test case.

### T3.2 - (Speed)

The purpose of this test is to verify that the results (counties) are returned by the back end and displayed on the map at an average of 5 seconds and at a maximum of 15 seconds. That it is reasonable for a user to acquire the information they choose within a reasonable amount of time which we will define as opening a browser, navigating the Code Fury Website and obtaining the requested information within two minutes.

#### Requirements addressed -- (Covers speed requirements)

Identified below is the CSCI or system requirements addressed by the test case. (Alternatively, this information may be provided in [5.a.)](#_41mghml)

Front to back testing, we will conduct unit tests on each individual feature to ensure time constraints are maintained, correctness of information will be addressed in a separate test case. Times on queries to our database will be recorded and analyzed resulting in proper performance. In addition to our database layout and intuitiveness will be considered by obtaining feedback from five volunteers.

A user should be able to successfully navigate the Code Fury Website with relative ease as features will be intuitive. Five volunteers with basic computer backgrounds will attempt to do so and provide feedback to developers. We will tell the volunteer that they will be able to search the United States with the prerequisite conditions listed below to find the most suitable place to live according to their specific input. If they wish, they may also change their input to see different results.

Timing for the user to load the webpage and achieve their goals will be recorded. For testing purposes we will ask the User to write down their goals prior to opening the web page. Feedback from this will help us streamline the layout in a better way so that users will easily navigate the web page and understand what our site provides.

#### Prerequisite conditions

The following considerations shall be discussed, as applicable:

* + - * 1. Hardware and software configuration

The user will be provided a working computer containing the Code Fury Webpage

* + - * 1. Flags, initial breakpoints, pointers, control parameters, or initial data to be set/reset prior to test commencement

Database will be maintained on Code Fury server that we provide via virtual machine

Database could be periodically updated should changes to our variables occur.

Functions that will query the Google API will be encapsulated and individually tested for correctness.

Functions that search our database will be encapsulated and individually tested for speed and accuracy.

* + - * 1. Preset hardware conditions or electrical states necessary to run the test case

Virtual Machine properly brought up with the packages described on our ReadMe.txt file easily seen from our Code Fury GitHub account.

* + - * 1. Initial conditions to be used in making timing measurements

We will not consider network delays during our testing.

Our network will consist of virtual machines which will will operate on a host only network.

* + - * 1. Conditioning of the simulated environment

Virtual machines in a host only network will ensure ideal environment which would be comparable to ideal network connection and minimal congestion over the internet.

* + - * 1. Other special conditions peculiar to the test case

none

#### Test inputs

This paragraph shall describe the test inputs necessary for the test case. The following shall be provided, as applicable:

1. Name, purpose, and description (e.g., range of values, accuracy) of each test input database will have all data for each county, queries will be made over the entire county if selection is of a particular state then that state alone will have results returned
   1. Outdoor activities
      1. query database and return boolean 0 or 1 if near the following activities
         1. biking
         2. climbing
         3. hiking
         4. hunting
         5. wilderness
         6. swimming
   2. Public School
      1. query database and return a normalized number between 0 and 1 for graduation rates
   3. Public Transportation
      1. query database and return boolean 0 or 1 if near transportation
   4. Climate Specifications
      1. query database and return counties which match the selected features as follows
         1. total snowfall per year
            1. low less than ten inches per year
            2. medium between 11 and 70 inches per year
            3. high over 71 inches per year
   5. Healthcare
      1. query database and return normalized data for doctors per resident for each county
   6. Average Commute
      1. query database and return the average commute times to job locations per county
2. Source of the test input and the method to be used for selecting the test input
   1. Source code for testing will consist of files labeled as unit tests in gitHub repository under Code Fury.
3. Whether the test input is real or simulated
   1. real
4. Time or event sequence of test input
   1. Execution times will be recorded, sequence will be described in a ReadMe.txt in the Unit Test folder in GitHub repository for Code Fury.
5. The manner in which the input data will be controlled to:

Test the item(s) with a minimum/reasonable number of data types and values

Exercise the item(s) with a range of valid data types and values that test for overload, saturation, and other "worst case" effects

Exercise the item(s) with invalid data types and values to test for appropriate handling of irregular inputs

Permit re-testing, if necessary

#### Expected test results

This paragraph shall identify all expected test results for the test case. Both intermediate and final test results shall be provided, as applicable.

We don’t have test cases yet, if we do have test cases and I have not updated this section please remind me to update this section. If you have reminded me and I have still not updated this section you have every right to punch me in the face. <dgess1>

#### Criteria for evaluating results

This paragraph shall identify the criteria to be used for evaluating the intermediate and final results of the test case. For each test result, the following information shall be provided, as applicable:

* + - * 1. The range or accuracy over which an output can vary and still be acceptable

We will tolerate a +/- 0.05 discrepancy in our data. Some cases may have special considerations due to the fact that we calculated median values of the k nearest neighbors.

* + - * 1. Minimum number of combinations or alternatives of input and output conditions that constitute an acceptable test result

One input will result in output of the top ten locations as determined by the input, Maximum input would consist of every possible parameter and will still result in top ten locations.

* + - * 1. Maximum/minimum allowable test duration, in terms of time or number of events

Building a database, if tested could take upto 10 hours depending on the specifications of testers hardware configuration, internet connection speeds and patience.

Searches of our database should take between a few seconds to one minute. Do not wait longer than a minute, you may kill the process and record a failure for that portion ofthe test. Recorded your results and notify a developer.

* + - * 1. Maximum number of interrupts, halts, or other system breaks that may occur

Will not be taken into consideration as this will vary depending on a great number of factors across individual testing environments.

* + - * 1. Allowable severity of processing errors

It’s a webpage, it will work or it will not. If you are experiencing any adverse processing errors perform a git pull to obtain an up to date package.

* + - * 1. Conditions under which the result is inconclusive and re-testing is to be performed

Should you obtain different results while searching under the same parameters, record your results and notify a developer.

* + - * 1. Conditions under which the outputs are to be interpreted as indicating irregularities in input test data, in the test database/data files, or in test procedures

Errors in our database: String offsets are incorrect, misspelling of names.

Errors in test data: Known samples do not equal the test results.

Errors in test procedures: Insufficient output in which to compare correctness with samples, or more output then required.

* + - * 1. Allowable indications of the control, status, and results of the test and the readiness for the next test case (may be output of auxiliary test software)
        2. Additional criteria not mentioned above.

more to follow

#### Test procedure

This paragraph shall define the test procedure for the test case. The test procedure shall be defined as a series of individually numbered steps listed sequentially in the order in which the steps are to be performed. For convenience in document maintenance, the test procedures may be included as an appendix and referenced in this paragraph. The appropriate level of detail in each test procedure depends on the type of software being tested. For some software, each keystroke may be a separate test procedure step; for most software, each step may include a logically related series of keystrokes or other actions. The appropriate level of detail is the level at which it is useful to specify expected results and compare them to actual results. The following shall be provided for each test procedure, as applicable:

1. Test operator actions and equipment operation required for each step, including commands, as applicable, to:
2. Initiate the test case and apply test inputs
3. Inspect test conditions
4. Perform interim evaluations of test results
5. Record data
6. Halt or interrupt the test case
7. Request data dumps or other aids, if needed
8. Modify the database/data files
9. Repeat the test case if unsuccessful
10. Apply alternate modes as required by the test case
11. Terminate the test case
12. Expected result and evaluation criteria for each step
13. If the test case addresses multiple requirements, identification of which test procedure step(s) address which requirements. (Alternatively, this information may be provided in [5.)](#_41mghml)
14. Actions to follow in the event of a program stop or indicated error, such as:
15. Recording of critical data from indicators for reference purposes
16. Halting or pausing time-sensitive test-support software and test apparatus
17. Collection of system and operator records of test results

1. Procedures to be used to reduce and analyze test results to accomplish the following, as applicable:

Detect whether an output has been produced

Identify media and location of data produced by the test case

Evaluate output as a basis for continuation of test sequence

Evaluate test output against required output

#### Assumptions and constraints

This paragraph shall identify any assumptions made and constraints or limitations imposed in the description of the test case due to system or test conditions, such as limitations on timing, interfaces, equipment, personnel, and database/data files. If waivers or exceptions to specified limits and parameters are approved, they shall be identified and this paragraph shall address their effects and impacts upon the test case.

# Requirements traceability

This paragraph shall contain:

1. Traceability from each test case in this STD to the system or CSCI requirements it addresses. If a test case addresses multiple requirements, traceability from each set of test procedure steps to the requirement(s) addressed. **(Alternatively, this traceability may be provided in** [**4.x.y.1.)**](#_1y810tw) **-- Try to do this so we can remove this section**
2. Traceability from each system or CSCI requirement covered by this STD to the test case(s) that address it. For CSCI testing, traceability from each CSCI requirement in the CSCI’s Software Requirements Specification (SRS) and associated Interface Requirements Specifications (IRSs). For system testing, traceability from each system requirement in the system’s System/Subsystem Specification (SSS) and associated IRSs. If a test case addresses multiple requirements, the traceability shall indicate the particular test procedure steps that address each requirement.

# Notes

This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.

# Appendixes

Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

DESCRIPTION/PURPOSE

The Software Test Description (STD) describes the test preparations, test cases, and test procedures to be used to perform qualification testing of a Computer Software Configuration Item (CSCI) or a software system or subsystem.

The STD enables the acquirer to assess the adequacy of the qualification testing to be performed.

APPLICATION/INTERRELATIONSHIP

Portions of this plan may be bound separately if this approach enhances their usability. Examples include plans for software configuration management and software quality assurance.

The Contract Data Requirements List (CDRL) should specify whether deliverable data are to be delivered on paper or electronic media; are to be in a given electronic form (such as ASCII, CALS, or compatible with a specified word processor or other support software); may be delivered in developer format rather than in the format specified herein; and may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document.

PREPARATION INSTRUCTIONS

General instructions.

a. Automated techniques. Use of automated techniques is encouraged. The term "document" in this means a collection of data regardless of its medium.

b. Alternate presentation styles. Diagrams, tables, matrices, and other presentation styles are acceptable substitutes for text when data required can be made more readable using these styles.

c. Title page or identifier. The document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem, or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of the preparing organization; and distribution statement. For data in a database or other alternative form, this information shall be included on external and internal labels or by equivalent identification methods.

d. Table of contents. The document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data in a database or other alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.

e. Page numbering/labeling. Each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data in a database or other alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.

f. Response to tailoring instructions. If a paragraph is tailored out of this document, the resulting document shall contain the corresponding paragraph number and title, followed by "This paragraph has been tailored out." For data in a database or other alternative form, this representation need occur only in the table of contents or equivalent.

g. Multiple paragraphs and subparagraphs. Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance readability.

h. Standard data descriptions. If a data description required by this document has been published in a standard data element dictionary specified in the contract, reference to an entry in that dictionary is preferred over including the description itself.

i. Substitution of existing documents. Commercial or other existing documents, including other project plans, may be substituted for all or part of the document if they contain the required data.