# CMSC 447
# Software Test Description (STD)

## Group 4 - */* No Comment*/*
## Realty Map Filter


Christopher Banos

Joseph Graso

David Udwin

Xavier Harris

Hannah Kiesel

Federico Cifuentes-Urtubey

# Table of Contents

# 1 Scope

## 1.1 Identification

This document applies to the program identified as the Realty Map Filtering (RMF) application. The version number as of May 9, 2018 is 1.0.0.

## 1.2 System overview

This application's purpose is to locate houses for sale across the United States (U.S.) that meet certain criteria defined by the user. These criteria include, but are not limited to, a location range, property data (e.g., market value), nearby schools, crime rates, and community types (e.g., urban, suburban, or rural). The results will be displayed on a Google Maps-style map using pins along with relevant details of the property listings.

This application will make requests to APIs from third-party solutions. We believe this will require security to prevent API results from being altered into false results. A possible method to avoid this is to use an HTTPS connection when the API is used.

As for personal information, our website will not be collecting any from the user, nor will we require a login. Therefore personal data should not be at risk.

# 2 Referenced documents

Software Development Plan (version 1), March 23, 2018.

Software Requirement Specification (version 1), March 28, 2018.

Software Design Description (version 1), April 8, 2018.

Software Test Report (version 1), May 1, 2018.

Software User Manual (version 1), May 13, 2018.

# 3 Test preparations

Considerations regarding safety, security, and privacy:

- Safety precautions -- WARNING: accurate map results are not guaranteed
- Security considerations -- NONE
- Privacy considerations -- NONE (user info and results are not stored)

## 3.1 (Project-unique identifier of a test)

### 3.1.1 Hardware preparation

There is no specific hardware requirement to run the RMF.

### 3.1.2 Software preparation

Required software to run the RMF:

- web browser with Javascript engine (e.g., Chrome, Firefox)
- Python 3.6+
- Flask
- flask-bower
- flask-triangle
- unicodecsv
- lxml
- beautifulsoup
- os
- requests
- fileinput
- glob

# 4 Test descriptions

This section shall describe test inputs, expected test results, criteria for evaluating test results, and test procedures.

Safety cautions:

- CAUTION: do not test on live version of software (if the software is being hosted on a website do not test on live version of website)
- CAUTION: run all significant changes by the other developers before pushing to live version.

## 4.1 Test inputs

Test inputs necessary for a test case:

| **description** | **source of test** | **real/simulated** | **min/max inputs OR input format** | **possible errors** |
|---|---|---|---|---|
| search bounds limited to US | manual input | real | | none |
| search radius between 1 - 500 miles | manual input | simulated | min: 1 miles<br>max: 500 miles | |
| properly formatted address | | simulated | | doesn't match pattern<br><br>incorrect spelling |
| address: street | | simulated | [0-9]{1,5}<br>[A-Z] or [a-z] | |
| address: city | | simulated | [A-Z] or [a-z] | |
| address: zip code | | simulated | [0-9]{5} [0-9]{4} | |

| | | | | |
|---|---|---|---|---|
| school | manual input | simulated | public, private | school doesn't exist |

## 4.2 Expected test results

Each requirement shall have a different test depending on which attribute of the search is being evaluated. See the following subsections for their descriptions.

## 4.3 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:

a. The range or accuracy over which an output can vary and still be acceptable

- search bounds limited to US
    - shall not vary
- search radius
    - must be greater than 0, however a number greater than 500 maybe acceptable as long as it does not go outside of US bounds
- properly formatted address
    - address must contain *at least* street, city, OR zip code
    - having all 3 at the same time is not necessary
- school
    - empty input is acceptable

b. Maximum/minimum allowable test duration, in terms of time or number of events

Percentages indicated amount of time/effort put into each test (does not add up to 100%)

- search bounds limited to US (80%)
- search radius (80%)
- properly formatted address (60%)
- school (20%)

c. Maximum number of interrupts, halts, or other system breaks that may occur

Max number of we are willing to accept is at least once a week. It should not be very difficult to restart this software when it crashes.

d. Allowable severity of processing errors

As long as it does not alter search results.

e. Conditions under which the result is inconclusive and re-testing is to be performed

Only condition in which this would happen is if there was no internet connection or the APIs were unreachable.

f. 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 search results were outside of the given bounds or do not match the given criteria, there is either a problem with the query sent to the API or a problem with the API itself.

### 4.4 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. The following shall be provided for each test procedure, as applicable.

- search bounds limited to US
    - set bounds to the outside of US borders
    - verify that the bounds reset to the border
- search radius
    - input “1” → accepts input
    - input “500” → accepts input
    - input “-50” → resets to “1”
    - input “1000” → resets to “500”
- properly formatted address
    - address must contain *at least* street, city, OR zip code
    - input “Sesame Street” → accepts input
    - input “40” → notifies user of invalid address
    - input “0803” → notifies user of invalid address
    - input “07052” → accepts zip code
- school
    - input “UMBC” → accepts input

## 5 Requirements traceability

In addition to the SRS requirement table describing traceability (SRS Section 3), this section defines how different components of the RMF will be testable to check if requirements are met.

a. map display

| Testable requirement | Passed/Failed | Date of Test |
|---|---|---|
| 1.0: search bounds limited to US | Passed | May 9, 2018 |
| 1.0.2: map radius is scalable | Passed | April 13, 2018 |

b. search inputs

| Testable requirement | Passed/Failed | Date of Test |
|---|---|---|
| 2.0.1: search criteria includes street address | Passed | May 9, 2018 |
| 2.0.2: search criteria includes city, state | Passed | May 9, 2018 |
| 2.0.3: search criteria includes zip code | Passed | May 9, 2018 |
| 3.0.6: search criteria includes square footage | Passed | May 9, 2018 |
| 4.0.6: search criteria includes school proximity | Pending | May 9, 2018 |

c. search results

| Testable requirement | Passed/Failed | Date of Test |
|---|---|---|
| 3.0.4: includes property listing cost | Pending | May 9, 2018 |
| 3.0.5: includes property type (e.g, condo) | Pending | May 9, 2018 |
| 3.0.6: includes property square footage | Pending | May 9, 2018 |
| 3.0.7: includes city, state of property listing | Pending | May 9, 2018 |
| 3.0.10: includes property number of bedrooms | Pending | May 9, 2018 |
| 3.0.11: includes schools nearby a property listing | Pending | May 9, 2018 |
| 3.0.13: includes crime rates regarding the property location | Pending | May 9, 2018 |

d. browser compatibility

| Testable requirement | Passed/Failed | Date of Test |
|---|---|---|
| 5.1.2: RMF is compatible with Google Chrome | Passed | May 9, 2018 |

# 6 Notes

Acronyms

API - Application Programming Interface

CSCI - Computer Software Configuration Item

HWCI - Hardware Critical Item

IDE - Integrated Development Environment

KPPX - Key Performance Parameter (number X)

RMF - Realty Map Filter

SDD - Software Design Description

SDP - Software Development Plan

SRS - Software Requirements Specification

STD - Software Test Description

STR - Software Test Report

SUM - Software User Manual

TLS - Transport Layer Security (protocol)

UI - User interface