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 Requirement Specification (final version), March 28, 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 (Python)
- lxml (Python)
- beautifulsoup (Python)
- os (Python)
- requests (Python)

- fileinput (Python)
- glob (Python)

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	N/A	None
Search radius is scalable	Manual input	Simulated	Min: 1 miles Max: within U.S. bounds	None
Properly formatted address		Simulated	See below.	Doesn't match input format Incorrect spelling
Address: street		Real	[0-9]{1,5} [A-Z] or [a-z]	
Address: city		Real	[A-Z] or [a-z]	
Address: zip code		Real	[0-9]{5}	
Property type	Manual input	Real	Checkbox (at least 1 must be checked)	Uncategorized property listing may appear
School rating	Manual input	Simulated	1-10	None
Price range	Manual input	Real	Min: < $100,000	None

			Max: $1,000,000+	

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 subsection shall identify the criteria to be used for evaluating the intermediate and final results of each 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 may be 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
   - price range
     - must be one of the selections from the drop menu

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

   The max number of interrupts, halts, or breaks observed on the system that is acceptable is three per day. If the application stops functioning correctly, it should be restarted as it is not very difficult to do so when it crashes.

d. Allowable severity of processing errors

   Processing errors are allowed 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 section shall define the test procedure for the each component of the RMF application. Each test procedure shall be defined by its input(s), which are independent of other search criteria. The pass criteria for each testable component is also given; tests for some components are automatically handled through how they are designed (i.e., dropdown menus are not negotiable and will always have an initialized selection). The following shall be provided for each test procedure, as applicable.

- search bounds limited to US
    - inputs require a zip code OR state, which are within US borders
    - **Pass criteria**: verify that the bounds reset to the border
- scalable map
    - Automatically allowed via the Google Maps API
- search radius
    - input “1” → accepts input
    - input “500” → accepts input
    - input “-50” → resets to “1”
    - input “1000” → resets to “500”
    - **Pass criteria**: must be an integer value greater than 0
- 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
    - **Pass criteria**:
        - a correctly spelled street name OR
        - a correctly spelled city name OR
        - a 5-digit zip code OR
        - any combination of the 3 above
- property type
    - **Pass criteria**: check one or more of the boxes (Townhouse, House, Condo)

- school rating
    - Automatically handled by the dropdown menu
- price range
    - Automatically handled by the dropdown menu

5 Requirements traceability

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

a. Map Display

Testable requirement
1.0: Search bounds are limited to US
1.0.1: Location of results are shown on a digital map
1.0.2: Map radius is scalable
4.0: The RMF shall utilize external interfaces to determine and display search results
4.0.2: Google Maps shall be used to display the map and geographic location of search results

b. Search Inputs

Testable requirement
2.0: Available user search options shall be able to filter through use of six search criteria
2.0.1: Search criteria includes street address
2.0.2: Search criteria includes city, state
2.0.3: Search criteria includes zip code
2.0.4: Search criteria includes neighborhood
2.0.5: Search criteria includes school
3.0.11: Search options shall include local schools

4.0.1: Redfin shall be used to to locate real estate addresses, prices, and, features of search results
4.0.3: SpotCrime shall be used to gather information about local crime rates to be used as a search option
4.0.5: SchoolDigger shall be used to gather information about school location to be used as a search option
4.0.6: search criteria includes school proximity

c. Search Results

Testable requirement
3.0: User search results shall display 26 results per page
3.0.1: Details of search results shall be displayed in a multi-page side window next to the map display
3.0.2: Available search results information shall be real estate to buy, rent, or just sold based on user search criteria
3.0.3: Results shall include zip code of real estate structure
3.0.4: Results include property listing cost
3.0.5: Results include property type (e.g, condo)
3.0.6: Results include property square footage
3.0.7: Results shall include house address
3.0.8: Results details shall include lot square footage
3.0.9: Results details shall include number of bathrooms
3.0.10: Results details shall include number of bedrooms
3.0.12: Results shall include local school ratings
3.0.13: Results shall include crime rates
3.1: Results shall be based upon a search from user inputs
4.0.4: SpotCrime shall be used to gather information about

local crime rates to be displayed as search results
4.0.7: SchoolDigger shall be used to gather information about school location to be displayed as search results
4.0.8: SchoolDigger shall be used to gather information about school ratings to be displayed as search results

d. Software Compatibility and Security

Testable requirement
5.0: The RMF shall ensure that users will not receive viruses, trojan horses or malware through the use of this application
5.1.1: RMF is compatible with Windows 7 through 10
5.1.2: RMF is compatible with Internet Explorer or Google Chrome

6 Notes

Acronyms

API - Application Programming Interface

IDE - Integrated Development Environment

RMF - Realty Map Filter

SDP - Software Development Plan

SRS - Software Requirements Specification

STD - Software Test Description

UI - User interface