

Executive Summary

Business Description

The business for which the project was performed is fictitious. However, all users of craigslist.org are the potential target market that will be served by the created project functionality.

System Overview

As users attempt to search for products on craigslist.org, they must search by only one city at a time. The “pain” here is evident in common use of the site as many users do not want to limit their search to one city, but desire to search cities nearby as well. Utilizing the craigslist.org Search Aggregator created in this project, users now have the functionality necessary to avoid manually researching nearby cities. They are now able to visually select each individual city to search via a form. The experience concludes with a selection of the most desired craigslist.org post and the form rendering that page visible to the user for full review.

General Form Path

The form makes use of many lists to aid users in a visual excursion to all of the desired cities they desire to search for a particular product. Steps are as follows:

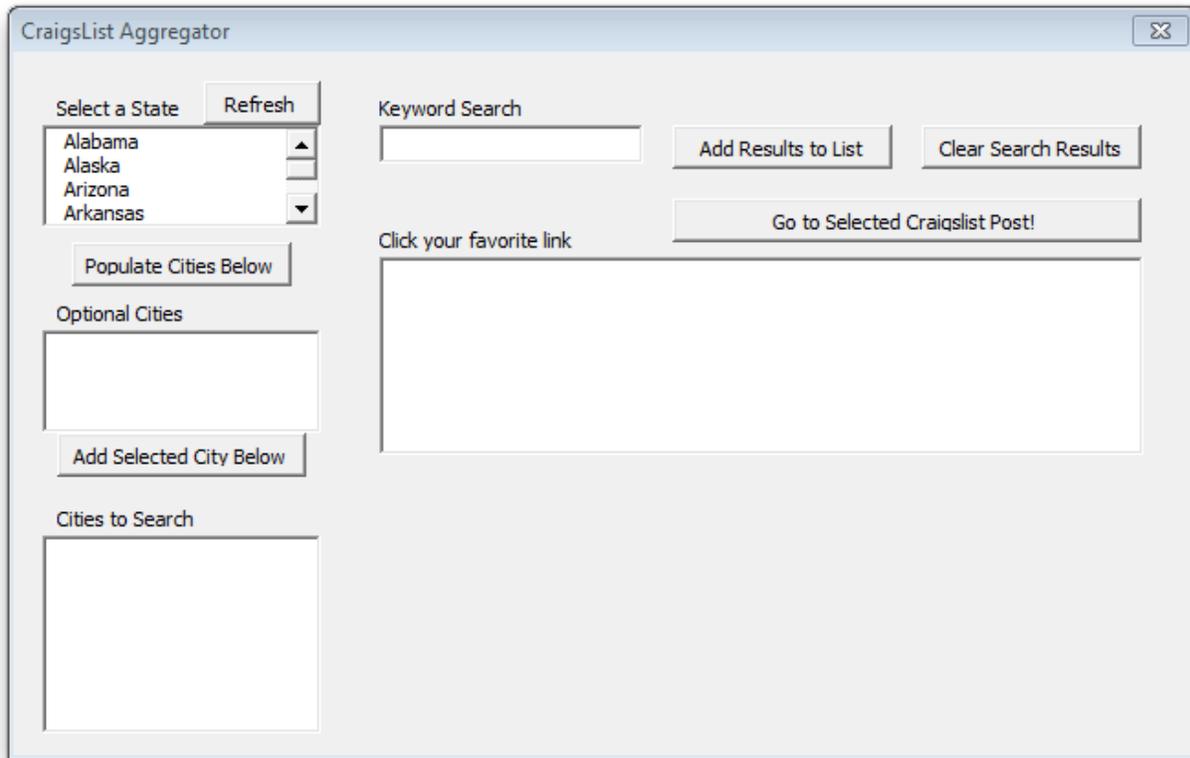
1. Selection of desired state
2. Selection of desired cities
3. Visual confirmation of each desired city
4. Users run the aggregated search
5. Selection of the desired post description
6. Viewing of the selected post in Internet Explorer

Optional Form Path Functionality

There is some functionality outside of the normal flow that adds considerable functionality to the experience. The Refresh button allows the “database” from which the form pulls data to be refreshed in case of the addition of new cities. It is important to note that the solution was created in a robust manner such that it will be a viable solution for craigslist.org users regardless of geographic availability changes made to the site. A “Clear” button was also added to allow for a clear post list and would allow for another product search to be commenced without need to reselect states and cities.

Implementation Documentation

Figure 1 will be referenced as each feature is explained by its purpose, description, and usage.



The screenshot shows a web application window titled "Craigslist Aggregator". On the left side, there is a "Select a State" dropdown menu with a "Refresh" button next to it. The dropdown menu is open, showing a list of states: Alabama, Alaska, Arizona, and Arkansas. Below the state list is a "Populate Cities Below" button. Underneath that is an "Optional Cities" text input field with an "Add Selected City Below" button. At the bottom left is a "Cities to Search" text input field. On the right side, there is a "Keyword Search" text input field with "Add Results to List" and "Clear Search Results" buttons. Below the search field is a "Click your favorite link" text input field with a "Go to Selected Craigslist Post!" button.

Figure 1 – Screenshot of user form

Features: “Select a State” List & “Refresh” Button

Purpose [the why]

The list is included so users may visually select their desired state and be able to refer back to it at any point. Knowing that at some point some cities will be added or deleted from the states, a “Refresh” button was added to update the cities for each state.

Description [the what]

The list is auto-populated from the sheet entitled “StatesAndCities” upon clicking the “Start” button on the “Start” sheet. It includes all states including Washington D.C. The “Refresh” button loops through the main craigslist.org state sites and parses the page for city information. Information is then saved on the sheet “StatesAndCities” under the column of the associated state.

Usage [the how]

The intended use of the list is to select a state in order to drill down into its cities for an eventual search of the city(ies) page(s). “Refresh” is intended to be used occasionally only when users feel that they want to be sure that all of the displayed cities will be current.

Features: “Optional Cities” List , “Populate Cities Below” Button, “Cities to Search” List

Purpose [the why]

The created form was designed to be simple and transparent to users about which information [state and city] is being used in the search. Users want to see step by step the hierarchal tree as it is being navigated from state to city. Also, included is the “Cities to Search” list which visually confirms the selection from all possible cities to search.

Description [the what]

Upon selection of a state and click of the “Populate Cities Below” button the state is searched in the “StatesAndCities” sheet. The associated cities are then pulled in from that column and added to the “Cities to Search” list as a visual confirmation.

Usage [the how]

It is intended that users select the state from above, click the button, and the respective cities are populated in the list for future selection.

Features: “Keyword to Search” Box , “Add Results to List” Button, “Clear Search Results” Button

Purpose [the why]

Both buttons were included to give users multiple uses from the same city/cities search. If the results list becomes too overwhelming or irrelevant at the time, users can clear the results and redefine the search.

Description [the what]

As users click the “Add Results to List” button, the “Keyword to Search” text box string is pulled from the form and compiled in a string. An agent instance scrapes the each city site for the string [product] and returns each found product to the “Craigs List” sheet. The link, description, and price are all saved to the sheet in separate columns. After being saved locally in the sheet, they are placed into the “Click Your Favorite Link” list and displayed to the user. A loop was used to iterate through each city listed in the “Cities to Search” list. The results of each subsequent city-product search are added to the bottom of the “Craigs List” page and the “Click Your Favorite Link” list is updated. The “Clear Search Results” button was added in case various products are desired for the set of selected cities. “Craigs List” is then cleared through deletion of the three columns and are then ready for another search.

Usage [the how]

Users are intended to review the cities selected, type a search criterion, and click the button in order to view the results that match the query in the list below.

Features: “Go to Selected Craigslist Post!” Button & “Click Your Favorite Link” List

Purpose [the why]

Users will undoubtedly want to be able to quickly navigate to the post that they feel is what they may want to read more about.

Description [the what]

Upon the button being clicked, users have selected a description in the list and an instance of Internet Explorer is opened for the user. Based on the description that is selected, the associated link is pulled from the sheet “Craigs List” and the agent class makes a visible instance of Internet Explorer open to the link.

Usage [the how]

Users are to select the desired post and click the button to view the post in the browser for more details.

Learning and Conceptual Difficulties

Beauty is in the Details

The “Refresh” button adds value to the form, but also takes substantial time. At the bottom of the window, a label was added that updates the user on the state that is being refreshed at the time. While it was added for user benefit, I found that even I was getting anxious for the module to finish because I could not discern the progress of the code. Looking back at the finished product, I believe that this is one of the most professional features that were included. I learned that users’ full acceptance of new technology will come from the extra features, and not only from core functionality. This skill will prove to be the difference between being good and being the best.

Easy Functionality, Hard Quality Assurance

Functionality is relatively easy to achieve, but extremely difficult to perfect. Automation provides opportunities to learn and expand your mind as you are forced to prepare your code for absolutely everything that it could encounter. As I built the module that parsed the web pages and inserted the descriptions into the spreadsheet, I found that seemingly random descriptions would break the code. How could a string be any different than the hundreds of others? After a visit to Brother Allen’s office, I found that every so often people would attempt to make their post more appealing and add a “====” before the words which Excel would interpret as a formula. I would have never thought that this could be a good problem. However, it trained my mind to search for every possible cause of technological issues and will aid in solving future problems in my VBA career. I know that I am a stronger thinker and programmer because of it and it ended as a positive experience.

Code Takes Longer Than You Think

Coding in theory seems easy to understand and implement. I quickly learned in this project in particular that it is easy to talk about the next needed feature, but the time hidden behind the words is much more than one would think. One instance in particular was when I pivoted from hard coding a single state’s cities to build basic search functionality to adding full city selection by the user. More than a quarter of the project work was spent on this seemingly easy addition of the next feature. Projects in coding are a procrastinator’s worst enemy and I learned a great lesson in working ahead.