

Automated Market Conditions Report
Final Project for MBA 614
Developed for Farr Appraising Services
By Blaine Farr

Executive Summary

For my final project, I decided to create an automated market conditions report taken from comparables listed in the database at www.UtahRealEstate.com. I designed this program specifically for my father's business, Farr Appraising Services. Currently, my dad prints the results from his search criteria and then uses a calculator to find the average prices for comparable homes sold in the past month, six months, and year. This process is slow, ineffective, time-consuming, and prone to errors in the calculation process.

When first run, the program asks for the user's username and password, and then automates the Internet Explorer Browser to log into the website database. At this point, a search criteria form is loaded that contains the options necessary for the search, including five multi-selection fields populated from the online options. Once populated, the search form is shown to the user who chooses and enters the criteria to search for comparables. From here, the data is used to fill in the search options on the online form and the online form is submitted.

Once the search results are available, the program automatically downloads the data into an Excel spreadsheet. The program then manipulates this spreadsheet to provide the important details on the comparables and to give the average prices for active and sold listings in the last 0-3 months, 4-6 months, and 6-12 months.

After running the program for the first time, the user is also able to repeat the process without reentering his or her username and password or reloading the search options. Another available option is to enter the information online and then continue running the program. These steps significantly cut down the processing time of the search and adds efficiency to the program.

Previously, my dad spent about 10-15 minutes performing this step for each of his appraisals. Assuming he does about 10 appraisals per week, this program could save him up to 100 hours per year, or two full weeks of work! This will allow him to spend more time with the family or to do more appraisals depending on the amount of work that is available.

Implementation Documentation

The main feature of the program consists of seven main subroutines: the main, login, loadForms, getData, submitData, createReport, and showResults subroutines. These subroutines have been modularized to create a code that is easier to read, execute, and troubleshoot. The following section will provide an overview of how these subroutines execute to run the program.

The Main Subroutine

This subroutine executes primarily by calling the other subroutines as seen in Figure 1. By using this as a main sub, the rest of the code is easier to read, reuse, and debug. The first line of code tests whether the search forms are currently loaded with the multi-select options. If the statusArray is empty, it means that (1) the user is not logged in, and/or (2) the search forms are not loaded. By creating this “If” statement, the program is able to skip the relatively slow login and loadForm procedures and proceed directly to inputting search criteria.

Figure 1—Main Sub

```
Sub getComps()  
  If statusArray.Count <> 0 Then  
    getData  
  Else  
    login  
    loadForms  
    getData  
  End If  
  
  submitData  
  createReport  
  showResults  
End Sub
```

The Login Subroutine

The login subroutine does exactly as it is entitled, it logs into the UtahRealEstate website. First, the code initializes and accesses the UtahRealEstate login page. An “On Error Go To” line of code allows the user to skip ahead if he or she is already logged in and the code returns an error. If not already logged in, the program displays a Login Form as seen in Figure 2 for the user to input his/her login credentials. This eliminates the need to hardcode this information, which is sensitive and changes every four months, into the code.

Figure 2—Login



The LoadForms Subroutine

Once the user logs in, the program automatically populates several fields on the search form: status, property type, style, area, and city. By automatically populating these lists on the search form, the code will stay up-to-date as these options change, will provide the means to submit these choices back into the server, and will prevent the user from entering information that is not valid. Because of the nature of the online HTML code, this step in the programming process was especially tricky and required a special program called Webshark and some help from Dr. Allen.

First, I found where the data came from in the HTML and wrote some code to copy and split the data into an array where each item had a unique value and text (ie. 502, Alpine). This step was not too difficult, but then required Dr. Allen to teach me about using collections of arrays to store data. After learning about collections of arrays, I split the data into an array of the value (0) and the name (1), and created a collection of these

arrays to hold information for all of the cities. With this I was able to write some code that populated the list box in the search form with the data from each array in the collection. I have included an example of this code in Figure 3 because I thought it was so interesting and useful.

The getData Subroutine

This subroutine displays the “Search Form” as seen in Exhibit 4 for the user to fill out and stores the inputted information for later use. I built the “Search Form” to be more user friendly than its online version by showing more of the multi-selection options, including a “reset” command that clears all of the data, and writing code to prevent some types of data entry errors, such as using a year before 1800 or a zipcode with more than five digits. As mentioned above, the list boxes are also automatically populated from the HTML code.

Once the user enters the search criteria and clicks “OK,” the program stores and manipulates the inputs in a format that is compatible with the online site. For the list boxes, this required me to create a string for the values associated with each city, style, property type, etc. To discover what type of format was required for the online form, I had to use the Webshark program and analyze the URL that is submitted when I made certain changes to the form. I was quite proud of this segment of code as it required an understanding of the array collection, array, and the format of the outputs to be compatible with the online form.

Figure 3—Array Collection Code

```
For x = 0 To UBound(cities)
    city = Split(cities(x), ":")
    statusArray.Add &_
        Array(city(1), city(0)), city(0)
Next

For x = 1 To statusArray.Count
    frmSearch.LstBoxStatus.AddItem &_
        statusArray(x)(0)
Next
```

Figure 4—Search Form

The submitData Subroutine

The submitData subroutine uses the data collected from the “Search Form” to submit the data onto the online search form. This process required a special approach of submitting a URL with the given search criteria because the layout and code of the website did not accept the normal processes.

To submit the data, the program passes the finalized inputs from the “Search Form” to another subroutine called addSomething with this line of code:

addSomething "param", value, "op"

The parameters are then encoded into a URL compatible form and put into this URL to update the online form that can be seen in Figure 5. The code to do this is below

"http://www.utahrealestate.com/search/chained.update/count/true/criteria/true? param=" & URLEncode(param) & "&value=" & URLEncode(val) & "&chain=criteriaAndCountAction" & " & op = " & URLEncode(op) & " &advanced_search=1&submit=submit"

This process is then repeated for each of the parameters and values until all have been entered. The program then submits the form to view the Search Results. From there, the program saves the search results to a local file and uses the Web Query Wizard to download the information into a worksheet.

Figure 5—Internet Search Page & Results Page

The screenshot shows the UtahRealEstate.com website in Internet Explorer. The browser address bar displays the URL: <http://www.utahrealestate.com/search/form/refine/1>. The website header includes the UtahRealEstate.com logo and navigation links: Home, Profile, Clients, Searches, CMA, Forms, Support, Tools, Statistics, Billing, and About Us. The main content area is titled "Searches / Residential / Full" and shows a search results summary: "Matches: 2,454". Below this, there are search filters and criteria. The "Search Criteria" section on the left lists: State: Utah, Status: Active or Sold or Under Contract, Property Type: Townhouse, Style: 2-Story, Area: Bntfl; NSL; Cntrl; WdX; Frmt, City: Abraham, Year Built: 1950, and Year Built: 1980. The "Search Criteria" section on the right includes: Status: Includes Active, Sold, Under Contract, List Price: 180000 to 200000, Property Type: Includes Townhouse, Style: Includes 2-Story, Number of Days Back: (empty field), Open House: No, Short Sale: Includes, Pending Price Approval: (empty checkbox), Price Previously Approved: (empty checkbox), SS Under 3rd Party Review: No, Location: Bntfl; NSL; Cntrl; WdX; Frmt, County: Choose County, State: Includes, and City: Includes. The bottom of the page shows a list of search results with columns: Location, Site, Details, Additional, Remarks, Listing, and Sold/Date. The taskbar at the bottom shows the Windows Start button and several open applications: Project Write up.d..., Microsoft Excel - T..., Microsoft Visual Ba..., and UtahRealEstate.com...

The data that is pulled into Excel comes from the following search results page seen in Figure 6 that I first save to a local file, and then direct the web browser to.

Figure 6—Search Results Page

UtahRealEstate.com | WFR Multiple Listing Service | Reports - Windows Internet Explorer

http://www.utahrealestate.com/search/perform

Google

File Edit View Favorites Tools Help

Google

Search

Share

Sidewiki

Bookmarks

Check

Sign In

UtahRealEstate.com | WFR Multiple Listing Service | R...

UtahRealEstate.com

Brought to you by Utah's Professional REALTORS®

Email Webmaster | Discussion Forums | Preferences | Help | Sign Out 59:56

Home Profile Clients Searches CMA Forms Support Tools Statistics Billing About Us

Searches / Residential / Results

Refine Search

Create Auto Hotsheet

Save Selections

Send Selections

New Search

Selected: 0 of 50

Report: Search Results

Prev

Page 1 of 1

Next

Page Preference

50 listings per pag

Search Criteria

State is Utah

Status is Active or Sold

Property Type is Single Family

Sold Price at least 90000

Sold Price at most 100000

List Price at least 90000

List Price at most 95000

ALL	Ref. #	Photo	MLS #	Address	List Price	Status	Style	City	State	Tot SqFt	Sold Price	NS	EW
			Bd Bth Fm	Gar Port	Bsmt %Fin		List DT	Sold DT	Acres	Year Bld	Type	Ph Cnt	
<input type="checkbox"/>	1		854157 35 W 3350 S	2/1.00/1 1/0	None/Crawl Space/0%	Sold	Bungalow/Cottage680	South Salt Lake, UT	UT	12/29/2008 04/22/2009	\$90,000 1951	0.17	3350 S 35 W Tour
<input type="checkbox"/>	2		891321 48 E 490 S	3/2.00/1 2/0	None/Crawl Space/0%	Sold	Manufactured/Modular1,512	Ivins, UT	UT	06/09/2009 07/21/2009	\$94,900 1997	0.14	490 S 48 E None
<input type="checkbox"/>	3		923444 93 E 500 S	2/1.00/1 0/0	Partial/0%	Sold	Bungalow/Cottage1,285	Nephi, UT	UT	11/02/2009 03/04/2010	\$90,000 1903	0.23	500 S 93 E None
<input type="checkbox"/>	4		893719 226 N 100 E	2/1.00/0 2/0	Partial/0%	Sold	Bungalow/Cottage1,142	Brigham City, UT	UT	06/17/2009 08/17/2009	\$98,200 1921	0.25	226 N 100 E None
<input type="checkbox"/>	5		922890 130 N 100 W	2/1.00/1 2/0	Partial/0%	Sold	Bungalow/Cottage2,298	Tooele, UT	UT	10/30/2009 01/26/2010	\$91,000 1925	0.17	130 N 100 W None
			930990			Sold	Bungalow/Cottage2,100				\$90,400		4400 S 174 E

Equal Housing Opportunity | Copyright © WFRMLS. All Rights Reserved. Information not guaranteed. Buyer to verify all information. Copyright Notice | Privacy Policy

6 items remaining

Downloading picture http://www.utahrealestate.com/images/site/balloon/balloon.png...

Internet

100%

start

Project Writ...

Microsoft Vi...

Test Report...

UtahRealEst...

UtahRealEst...

UtahRealEst...

100%

11:23 PM

The createReport Subroutine

After downloading the information into the “LocalQuery” worksheet, the main Sub calls the createReport sub to organize the data into an understandable report. Compared to the rest of the code that works with the URLs and HTML code, this part of the program was extremely simple to write.

In this section, the code first runs several arrays to gather the necessary information for each criterion. It then stores these values in the worksheet called “Report.” In addition to creating arrays, assigning values to those areas, and then displaying the area in a different area, I also had to create several new arrays. For example, I created the Days on Market Array that calculates how long an Active Listing has been on the market or how long a Sold Listing had been on the market. The full report is shown in Figure 7.

Figure 7—Detailed Report Page

MLS	Address	City	State	Status	Type	Style	List DT	List Price	Days on M	Sold DT	Sold Pr	Tot Sq	Year Bld	Acres	Bld/Bth/L	Gar/P
82141	2105 PROSPECTOR AVE 310	Park City, UT	UT	Active	Condo 12	Other/See Remarks	7/31/2008	\$90,000	621			315	1978	0	0/1/0/0	0/0
938493	410 S LUCERNE DR	Salem, UT	UT	Under Contr	Single Family 11	Rambler/Ranch	1/29/2010	\$164,300	74			2250	1975	0.19	42.00/0	0/1
938022	4941 S VERNAL AVE	Utah County, UT	UT	Under Contr	Single Family 18	Rambler/Ranch	2/10/2010	\$170,000	71			1175	1978	1.15	32.00/1	0/2
938985	1834 S ROCKCRESS DR.	St. George, UT	UT	Active	Single Family 12	Other/See Remarks	2/14/2010	\$3,800,000	68			7520	2004	1.38	47.00/1	365
93974	5658 OSLO LN	Park City, UT	UT	Active	Townhouse 12	Townhouse/Row-m	2/14/2010	\$473,000	68			2225	2005	0.01	44.00/2	365
938228	9010 UPPER LANDO LN	Park City, UT	UT	Active	Single Family 12	Other/See Remarks	2/19/2010	\$843,000	71			5780	1998	0.34	5/5.00/2	365
936610	7705 BUCKBOARD DR	Park City, UT	UT	Active	Single Family 12	Other/See Remarks	1/19/2010	\$693,900	84			4245	1992	0.5	54.00/1	365
940755	2095 S MAIN ST #9	Bountiful, UT	UT	Under Contr	Condo 13	Condo, Top Level	2/6/2010	\$134,300	66			931	1995	0.02	2/2.00/1	0/1
923109	137 SAMAK COUNTRY EST	Kamas, UT	UT	Active	Single Family 14	Cabin	9/6/2008	\$293,300	615			1540	1979	1.47	91.00/1	0/0
938526	545 S VALLEY VIEW DR 51	St. George, UT	UT	Expired	Single Family 18	Other/See Remarks	1/29/2010	\$239,300	74			1936	1997	0.12	3/2.00/0	365
933265	1700 THREE KINGS DR 182	Park City, UT	UT	Under Contr	Townhouse 12	Townhouse/Row-m	2/2/2010	\$650,000	70			1892	1972	0	4/3.00/2	0/0
938698	1985 EMPIRE AVE	Park City, UT	UT	Expired	Condo 12	Other/See Remarks	2/12/2010	\$643,000	69			1848	1991	0.1	3/3.00/1	365
942009	3011 N MAIN ST	London, UT	UT	Off Market	Single Family 12	Rambler/Ranch	2/12/2010	\$259,300	60			3574	1960	0.58	4/3.00/2	366
924487	289 S MAIN	Kanab, UT	UT	Expired	Single Family 12	Bungalow/Cottage	9/13/2008	\$95,000	608			1371	1900	0.63	3/1.00/0	0/0
942629	490 N STATE ST	Mount Pleasant, UT	UT	Active	Single Family 12	2-Story	9/12/2008	\$199,500	609			2760	1914	0.58	5/2.00/0	402
942466	381 HUNTINGTON HILLS RD	Vashington, UT	UT	Active	Single Family 10	Rambler/Ranch	2/16/2010	\$129,000	56			1170	1978	0.21	3/1.00/1	0/2
941976	1940 MAHRE DR	Park City, UT	UT	Active	Single Family 11	Tri/Multi-Level	2/12/2010	\$1175,000	60			4356	1998	0.3	44.00/1	365
940246	684 BUENA VISTA BLVD #604	Vashington, UT	UT	Active	Single Family 13	Other/See Remarks	2/5/2010	\$70,000	67			1088	1996	0.02	2/2.00/1	0/0
926476	381 S STATE ST	Levan, UT	UT	Active	Single Family 11	Rambler/Ranch	8/20/2008	\$163,300	601			2149	1978	1	3/3.00/1	0/0
931951	1909 1/2 MAIN ST	Bountiful, UT	UT	Active	Condo 12	Rambler/Ranch	2/10/2010	\$129,300	71			1248	1977	0.01	3/2.00/0	0/2
941021	1101 S CARBON AVE #10	Price, UT	UT	Active	Mobile (w/o Land) 11	Mobile	2/19/2010	\$15,375	63			980	1974	0.06	2/1.00/1	0/2
941019	733 DUTCH CREEK CT	Midway, UT	UT	Active	Single Family 12	Rambler/Ranch	2/19/2010	\$647,000	63			5440	2002	0.57	44.00/2	365
941657	2429 HOLIDAY RANCH LOOP RD	Park City, UT	UT	Active	Single Family 18	Cabin	2/11/2010	\$693,000	61			1500	1977	0.3	3/2.00/0	365
937952	119 S STATE ST	Richmond, UT	UT	Active	Single Family 12	Bungalow/Cottage	1/27/2010	\$142,500	76			2178	1939	0.27	3/1.00/1	365
939369	9811 N TRAILS DR	Park City, UT	UT	Active	Single Family 11	Tri/Multi-Level	2/2/2010	\$1243,000	70			6163	2006	1.87	67.00/3	365
939870	1536 N CAMBRIDGE AVE	Vashington, UT	UT	Active	Single Family 12	Other/See Remarks	1/20/2010	\$3,600,000	83			9749	2010	0.52	5/7.00/2	365
931955	151 PARK AVE	Park City, UT	UT	Active	Single Family 12	Other/See Remarks	3/11/2008	\$1,380,000	579			2208	1980	0.04	3/3.00/1	365
932318	16800 SUMMIT DR	Daniels, UT	UT	Expired	Single Family 12	Cabin	9/12/2008	\$1,388,888	578			7068	2005	1.43	6/6.00/3	366
939683	3161 SADDLEBACK RDG	Park City, UT	UT	Withdrawn	Single Family 125	Rambler/Ranch	2/10/2010	\$2,695,000	71			6395	2006	1.01	5/7.00/3	0/0
939512	7300 N HIGHWAY 38	Honeyville, UT	UT	Under Contr	Single Family 11	Rambler/Ranch	2/10/2010	\$197,000	69			2310	1951	1	4/2.00/2	365
933176	87 YAMAHCA CT	Park City, UT	UT	Expired	Condo 11	2-Story	3/15/2008	\$393,000	575			1430	1976	0.01	3/4.00/1	365
940708	4337 BOULDER VIEW DR 103	Hurricane, UT	UT	Active	Single Family 18	Other/See Remarks	2/18/2010	\$490,000	64			2300	2007	0.2	3/3.00/1	0/0
933512	1211 COTTONWOOD CT	Heber City, UT	UT	Expired	Single Family 18	2-Story	9/17/2008	\$265,000	573			2509	1997	1.3	3/3.00/1	365
939354	975 N HORSEMAN'S PARK DR	Dammoner Valley, UT	UT	Active	Single Family 10	Other/See Remarks	2/2/2010	\$489,300	70			3230	2005	2.81	4/0.00/1	365
936947	0 SEE REMARKS	Summit County, UT	UT	Active	Recreational 16	Cabin	9/25/2008	\$843,300	565			2398	2000	6.63	3/2.00/1	365
937694	3743 MULBERRY DR	St. George, UT	UT	Active	Single Family 12	2-Story	1/27/2010	\$479,000	76			2400	1976	6.78	4/3.00/1	365
939350	3189 11TH FAIRWAY	Vashington, UT	UT	Active	Single Family 11	Other/See Remarks	2/2/2010	\$181,900	70			1160	2004	0.1	3/2.00/1	365
936341	1940 PROSPECTOR AVE 317	Park City, UT	UT	Expired	Condo 15	Other/See Remarks	9/26/2008	\$125,500	564			490	1984	0	91.00/1	0/0

* Note: Sold Date and Sold Price are empty because these listings have not yet been sold.

The showResults Subroutine

The final call from the main sub refers to the showResults subroutine that is used to show the results in the table on the main worksheet. The purpose behind this subroutine is to provide the user with the information that he or she needs for the appraisal.

To make the necessary calculations, I embedded a “Select Case” Construction within an “If” Construction. This enabled me to screen through the statuses for the “sold” or “active” status with the “If” construction, and then to find the number, average days on market, and average price for comparables that have been listed or sold within the past three months, past four to six months, and past year respectively. This is best explained by the code and the actual worksheet as seen below in Figures 8 and 9.

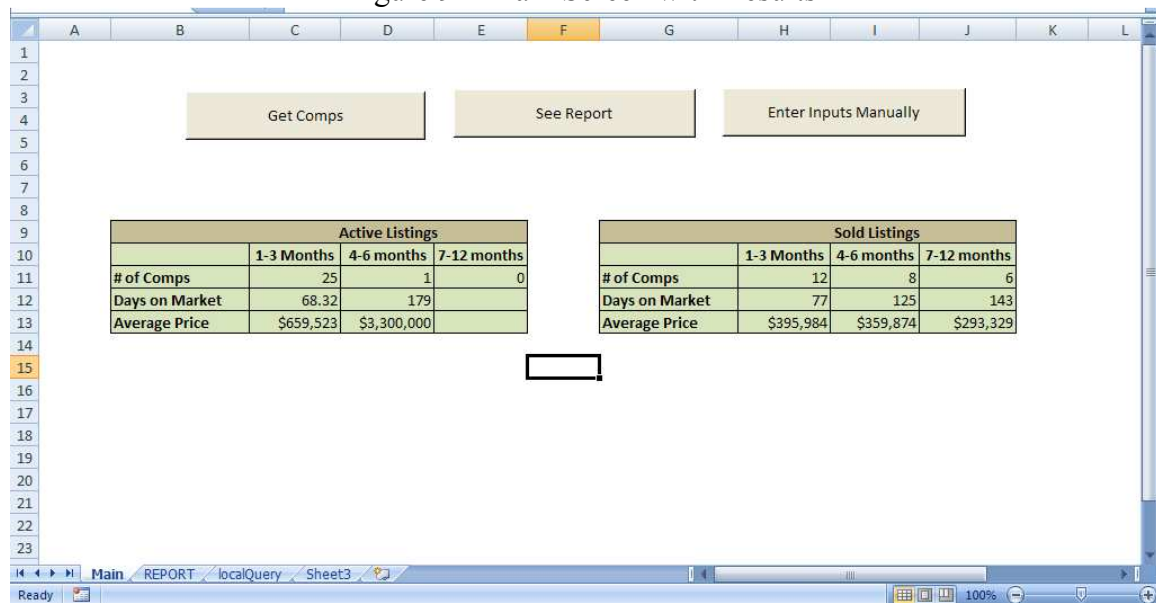
Figure 8—Code to Calculate Results

```

For x = 0 To count
    If Sheets("report").Cells(x + 2, 4).Text = "Sold" Then
        Select Case Sheets("report").Cells(x + 2, 9).value
            Case Is <= 91
                threemonth = threemonth + 1
                threemonthavg = threemonthavg + Sheets("report").Cells(x + 2, 9).value
                threemonthprice = threemonthprice + Sheets("report").Cells(x + 2, 11).value
            Case 91 To 180
                sixmonth = sixmonth + 1
                sixmonthavg = sixmonthavg + Sheets("report").Cells(x + 2, 9).value
                sixmonthprice = sixmonthprice + Sheets("report").Cells(x + 2, 11).value
            Case 183 To 365
                oneyear = oneyear + 1
                oneyearavg = oneyearavg + Sheets("report").Cells(x + 2, 9).value
                yearprice = yearprice + Sheets("report").Cells(x + 2, 11).value
        End Select
    End If
Next

```

Figure 9—Main Screen with Results



Additional Subroutines

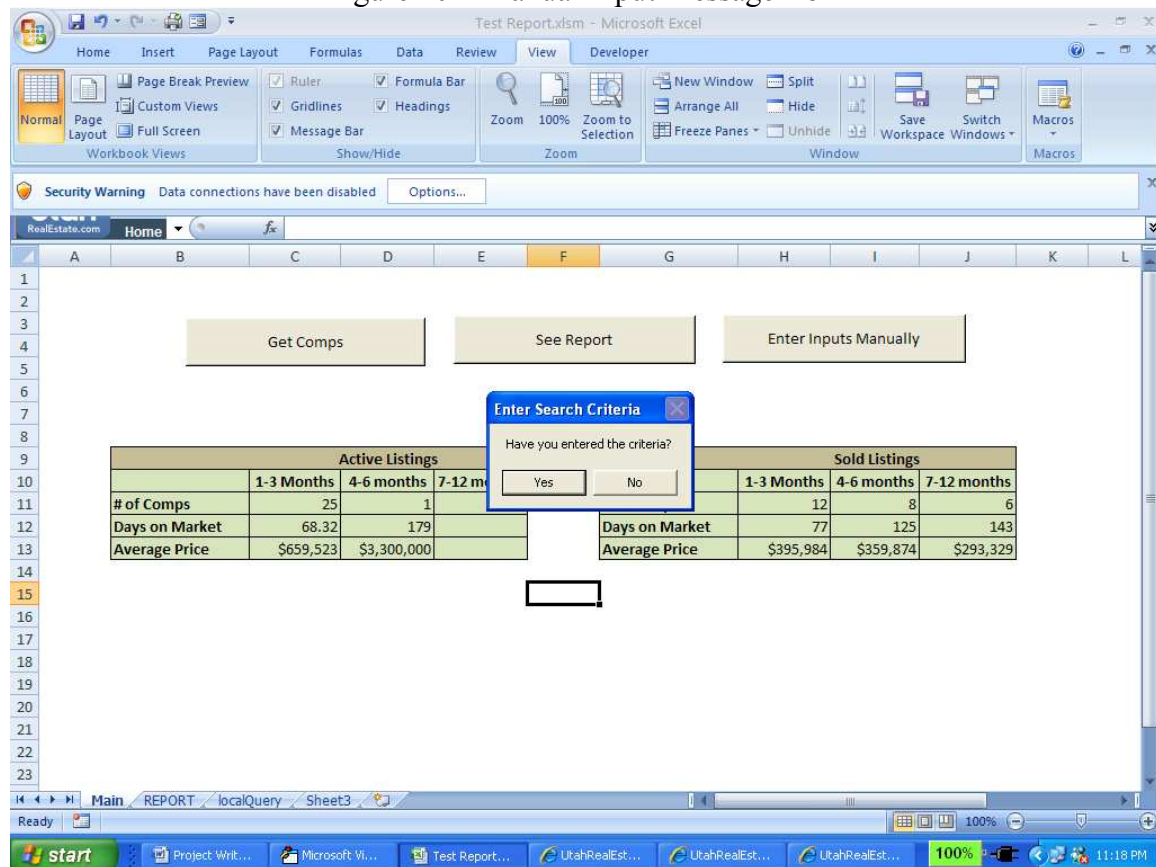
After finishing the program, I felt that I could still make the program faster, more user-friendly, and more error proof. To do this, I created another command button called “Enter Inputs Manually.” Because I modularize my program as seen in Figure 10, this sub was extremely easy to create and works very effeciently with fewer errors than the rest of my program because it is simply less complex. In fact, I believe that this feature will be the most useful for my father because it is less likely to have problems either with being installed on his machine or with his limited knowledge of Excel.

Figure 10—Manual Input Code

```
Sub enterManually()
    initializeIE True
    login
    ie.navigate "http://www.utahrealestate.com/search/form/type/1/name/"
    result = MsgBox("Have you entered the criteria?", vbYesNo, "Enter Search Criteria")
    If result = vbYes Then
        downloadData
        createReport
        showResults
    End If
End Sub
```

This code operates by first logging into the UtahRealEstate Website, just like the other code. It differs, however, in that it goes directly to the Online Search Form and pulls it up for the user. A message box then appears telling the user to enter their search criteria on the browser. Once they do this, they return to the Excel Program and click “Yes” as seen in Figure 11.

Figure 10—Manual Input Message Box



Note: After submitting my project, I also added a “Clear Results” command that simply clears all of the results that are being shown so that they are not confused with the next subject appraisal.

This program runs in about 1/10 of the time of the full program, or maybe faster, and also allows the user to see how narrow their search results have become. Once they click “Yes,” it runs the same code as the full getComps subroutine. I believe that this improvement will be extremely helpful for my dad because it is easier to fix, easier to understand, faster, and less prone to bugs and human errors, of which I have had many while working with the HTML, URLs, and so much more.

Learning and Conceptual Difficulties

I learned an incredible amount of new programming skills while working on this project. Coming into the class, I did not have any VBA programming experience, but I have worked hard to stay up with the class and feel confident in my ability to tackle beginning to intermediate-level work. This project, however, required some very advanced programming and experience in HTML and the learning curve was steep! Among the most important lessons, I learned how to approach and tackle extremely difficult programming problems and where to find answers to my questions. This helped me gain experience in HTML, solidify my understanding of VBA, and improve my problem-solving skills.

The first thing I learned was how to control and manipulate the Internet Explorer browser. This task turned out to be more difficult than I expected because the design of the website required me to learn some HTML code and differed from the standard online forms. In relation to this, I also learned how to use Wireshark, a program that reads incoming and outgoing packets. This taught me new methods to find answers to my questions.

The next major lesson that I learned is how to build userforms and automatically populate listboxes within those forms. I also learned how to capture this information to be compatible with its end use or requirement. One of the challenges in doing this was learning about and using collections of arrays. Before this project, I had never even heard of array collections, but this tool was essential to finishing this project.

I also learned how to download necessary data from an HTML site by saving the source code to a local file, and then referring to that code instead of the HTML. I feel that this lesson will be very useful in the future and should be taught to future classes. In addition, I learned how to manipulate data and create reports from the downloaded information. Doing so raised a few challenges because the data that I needed came in rows of three with the unique identifier on the first row. I was able to create arrays that solved this problem using “If” constructions.

Finally, I learned that by reviewing my work after I have completed it, I can find even better ways to write the program to make it run faster, more effectively, or more efficiently. This was a great lesson and I am very proud of my final project, and look forward to presenting it to my dad this week.

Conclusion

In conclusion, I have truly learned a lot from this project. I will be returning home this next weekend and look forward to showing it to my father. Depending on the amount of work he has in a given week, this program could save him several hours doing “busy work” that is much better automated. Those several hours each week will save him thousands of dollars over the course of a year. I believe that this project will be very valuable for him and his company.