Executive Summary

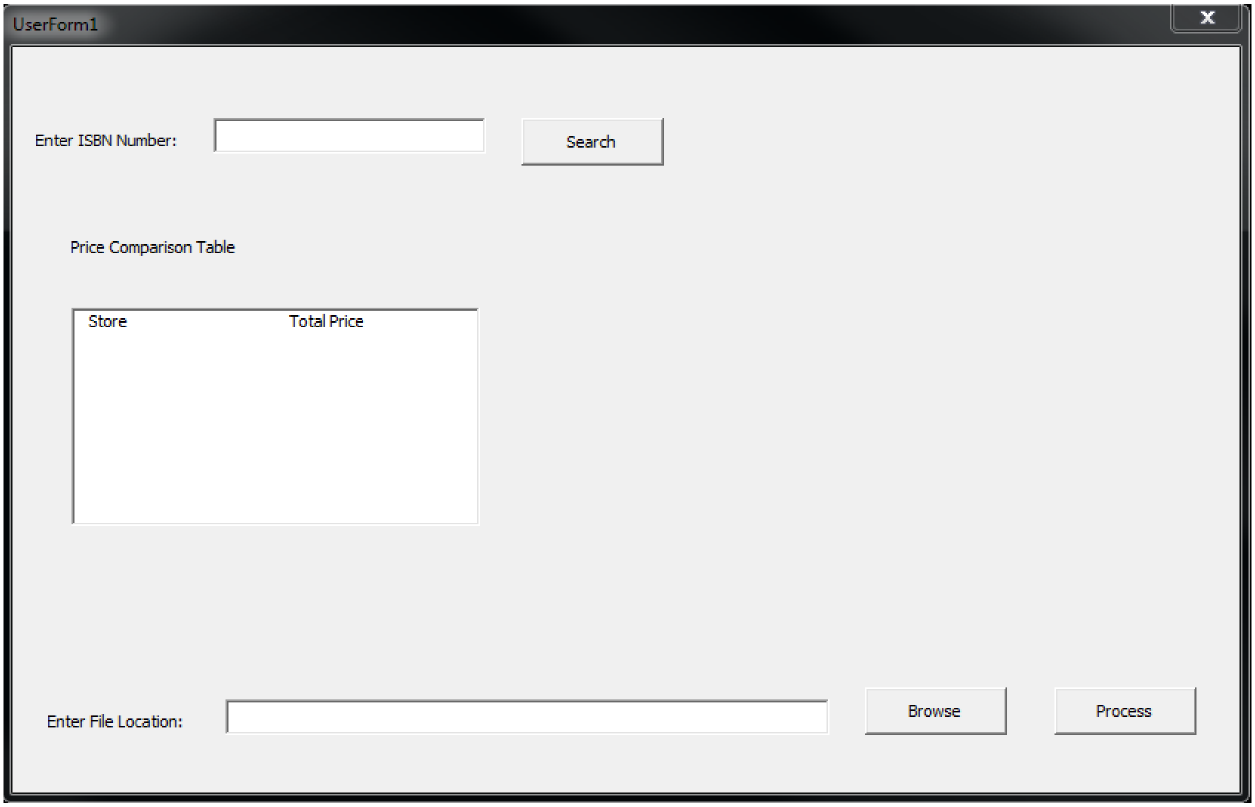
Every beginning of the semester, I would go through my book list and make search to different websites in order to find a lowest price for every book I was required to purchase. This was a very tedious process, copying ISBN number from your book list and then make search to different websites. Not anymore. My VBA project allows a user to upload BYU book list (a saved html page of BYU book list) and it will parse all the ISBN numbers in the book list and make a web search in order to find a lowest price for that particular book. At last it will write all different store prices for all books in the list into an excel file. The program will also allow user to select a destination folder for output excel file. Apart from this my program will allow user to make a single search by Isbn Number.

Implementation

I have a user form that will allow users to make a quick ISBN search for prices or upload a book list to get an excel file with prices from different stores as an output. Here are the steps; a user needs to follow in order use this project.

Step 1: Run the program.

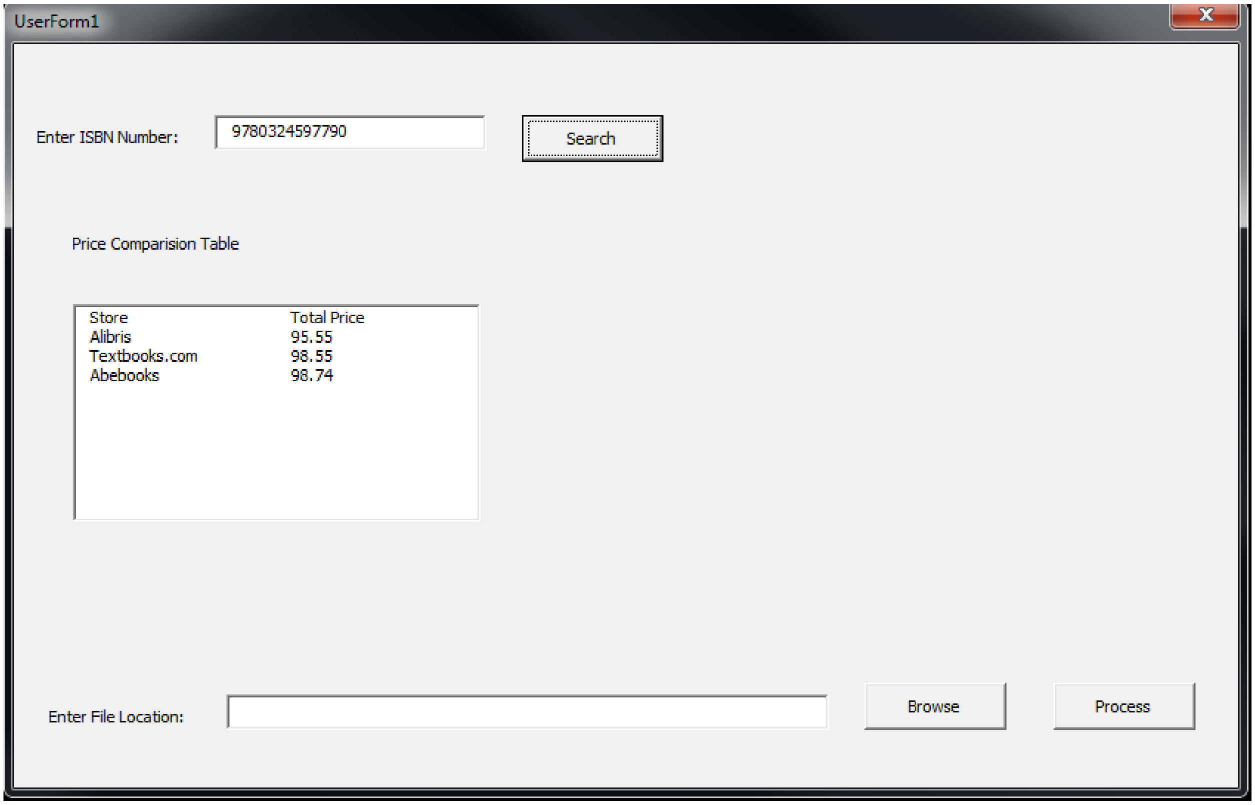
When you run the program you will get a user form. The screen shot of the user form is given below.



Step 2: Make a quick ISBN search

In order to make a quick single ISBN search, you have to enter ISBN number in the ISBN text field, the text field with search button on its right. After entering the ISBN number, you have to click search button.

When you click search button, you will get the prices from different stores in the list box present at the left center of the user form. This is how it will look like.

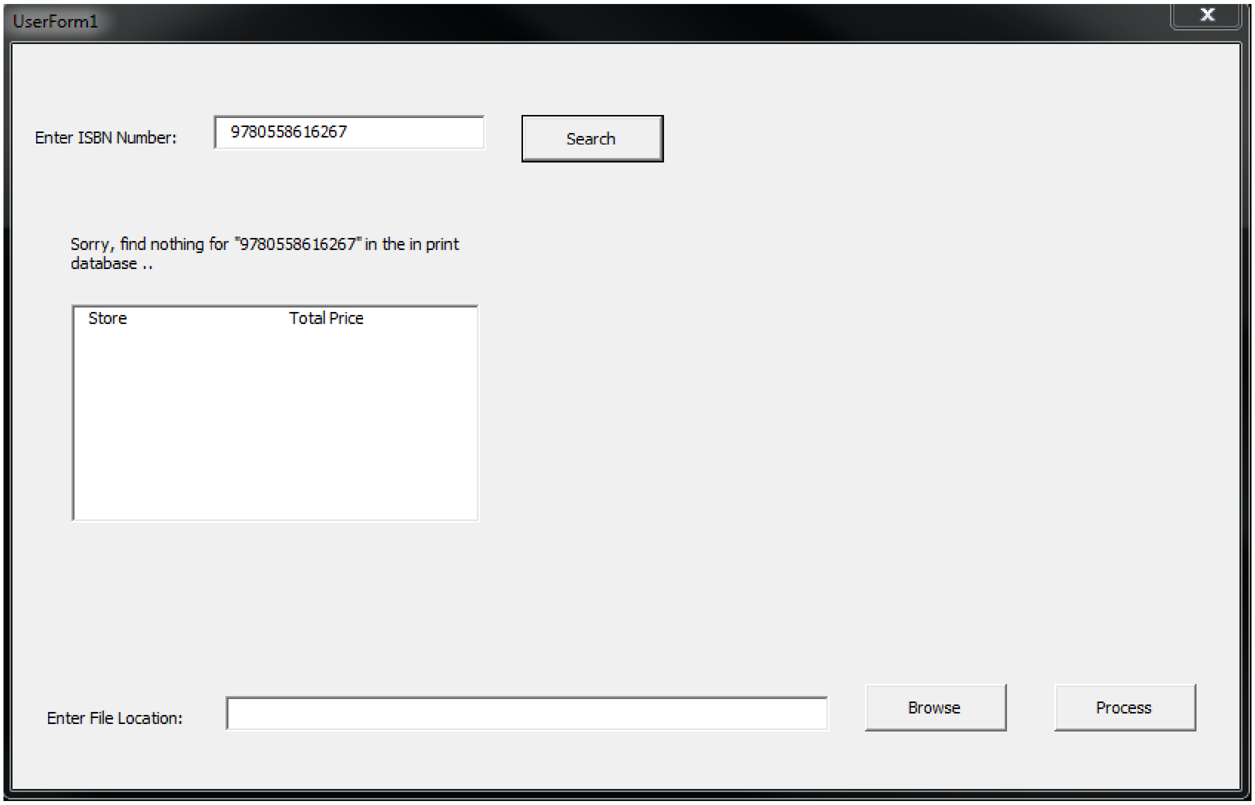


Enter ISBN Number: 9780324597790

Store	Total Price
Alibris	95.55
Textbooks.com	98.55
Abebooks	98.74

On the above process, whenever the user click search button my program will do a web query to a website (addall.com) and make a search for the given ISBN number. I will import the resulting webpage to an excel sheet and grab all the required information and display it in a multiple column list.

If a user make a search for a book that is not available in the market, for e.g. study packet from BYU professor, the program will not give you a list of books but instead it will give you a notification. The screen-shoot looks like this.



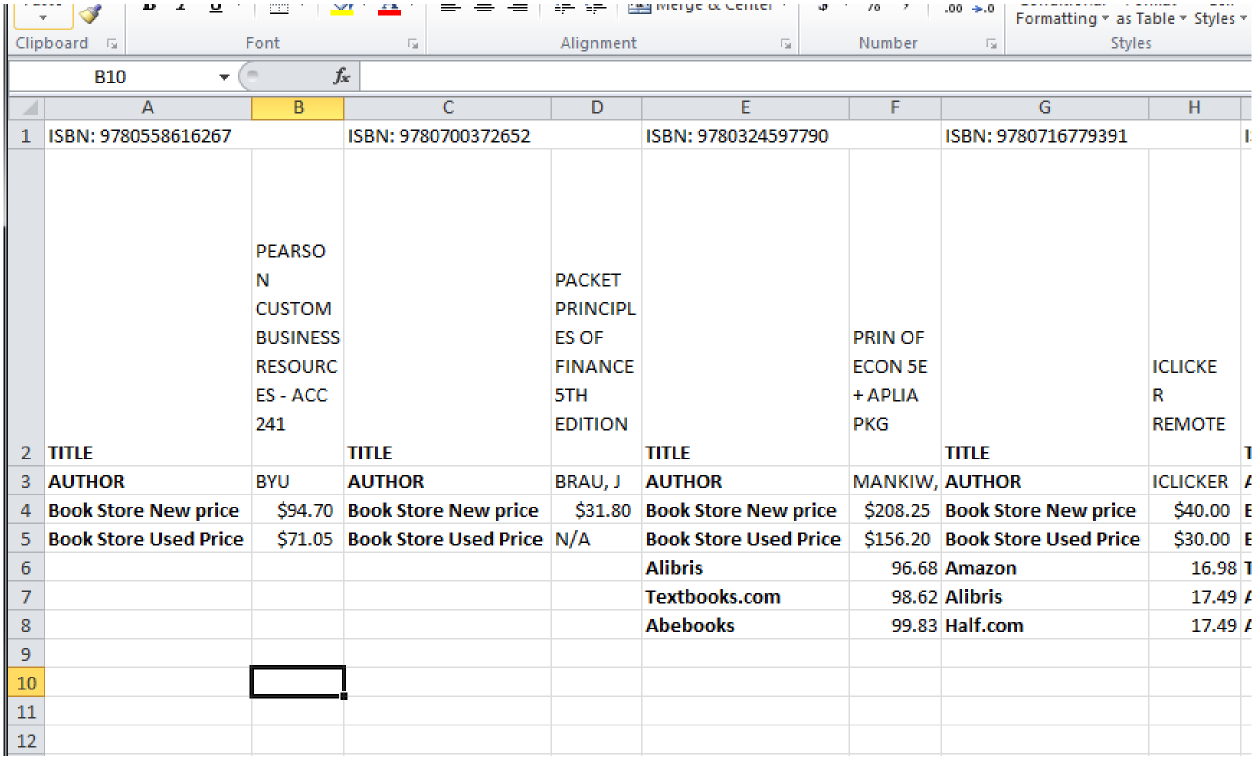
Enter ISBN Number: 9780558616267

Sorry, find nothing for "9780558616267" in the in print database ..

Enter File Location:

Step 3: Uploading a book list and getting an excel file as an output

This is the core feature of my project. In order to use this feature, first you have to save your BYU book list page in your local machine. After you save you book list, click on the browse button. This will open a file chooser dialog box. Navigate to your saved book list page; select it before you click ok. After you click ok, you will see the file path in the file location text box. Instead of browsing to your saved book list you could also enter the file path in the text box directly. After doing this you have to click on the process button. After some time, it will open a file saving dialog box. This will allow you to select the destination and name of the output file. The output file will look like this.



	A	B	C	D	E	F	G	H	I
1	ISBN: 9780558616267		ISBN: 9780700372652		ISBN: 9780324597790		ISBN: 9780716779391		I
2	**TITLE**	PEARSON CUSTOM BUSINESS RESOURCES - ACC 241	**TITLE**	PACKET PRINCIPLES OF FINANCE 5TH EDITION	**TITLE**	PRIN OF ECON 5E + APLIA PKG	**TITLE**	ICLICKER REMOTE	T
3	**AUTHOR**	BYU	**AUTHOR**	BRAU, J	**AUTHOR**	MANKIW,	**AUTHOR**	ICLICKER	A
4	**Book Store New price**	$94.70	**Book Store New price**	$31.80	**Book Store New price**	$208.25	**Book Store New price**	$40.00	E
5	**Book Store Used Price**	$71.05	**Book Store Used Price**	N/A	**Book Store Used Price**	$156.20	**Book Store Used Price**	$30.00	E
6					**Alibris**	96.68	**Amazon**	16.98	T
7					**Textbooks.com**	98.62	**Alibris**	17.49	A
8					**Abebooks**	99.83	**Half.com**	17.49	A
9									
10									
11									
12									

In the above process, when the user select the book list file and click process, first the program will parse the html content of the book list and extract ISBN number, Title, Author, Book Store New price, and Book store Used price of all books. For every book the program will repeat step 2 (i.e go to the addall.com and make a search for a given book). The program will open an excel file and write all the information scraped from the html book list and website (addall.com) into it. The resulting output looks like the above picture.

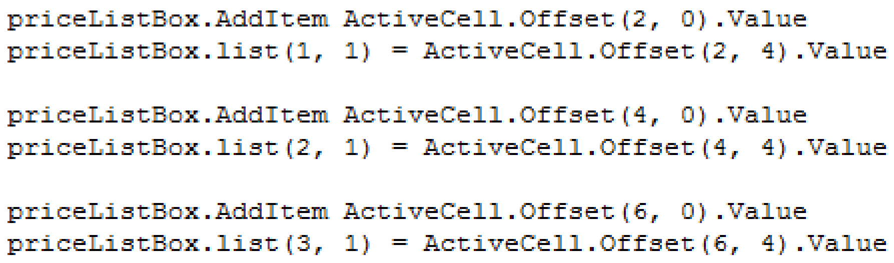
Learning and difficulties

While doing this project I struggled with multi-column list box. After doing some Google search, I found the trick to display my information in right format. At first, whenever I tried to add a new item to the list it would create a new row and display that item. But that is not what I wanted; I wanted to display my every second item in the second column not second column. Here is how my code looked at first,

```
priceListBox.AddItem ActiveCell.Offset(2, 0).Value
priceListBox.AddItem ActiveCell.Offset(2, 4).Value
priceListBox.AddItem ActiveCell.Offset(4, 0).Value
priceListBox.AddItem ActiveCell.Offset(4, 4).Value
```

This way I was creating a new row for every item I added to the list.

And the trick to make it right was,



```
priceListBox.AddItem ActiveCell.Offset(2, 0).Value
priceListBox.list(1, 1) = ActiveCell.Offset(2, 4).Value

priceListBox.AddItem ActiveCell.Offset(4, 0).Value
priceListBox.list(2, 1) = ActiveCell.Offset(4, 4).Value

priceListBox.AddItem ActiveCell.Offset(6, 0).Value
priceListBox.list(3, 1) = ActiveCell.Offset(6, 4).Value
```

In this code, after adding a new item to the list, which creates a new row in the list, i am inserting the price value in the second column specifically. If you have three columns list box then you will do the same thing, you will add one more line of code for each row in which you will specify the value to go into the third column.

Second hardest thing was parsing the html file. Firs I had to get the chunk of html that has all the information for one book. And again I had to find specific information within the first extracted html. I had to use two for loops in order to get all the information. While doing this the difficult part was to extract the right information and ignore other.

This project is exactly what I wanted it to be. I don’t have any elements that I wanted to include but couldn’t. This might not be the best way to do but this is exactly what I wanted to do. If any of you have a suggestion to make it better, I am open to make any changes to this project.

Conclusion

Wow, it feels good when you are done with something that you wanted to for a long time. I am happy the way this project turned out. Ever since my first semester in BYU, this is exactly what I wanted to do. Now I will be saving lot of time and money. I hope this project will be useful to other BYU students as well.