Kurt Hansen: VBA Final Project

Executive Summary

My VBA project establishes a database for piano teachers, their students and the students' repertoire in 5 different musical periods: Baroque, Classical, Romantic, Impressionist and 20th Century. This database includes all details of a pianist's entire repertoire, including era, composer, the piece title (and any accompanying nickname), and the catalogue number if applicable (i.e. "Opus" or "BWV #", etc.).

When the form "PianoRepertoireForm" is opened, the user first selects a teacher listed in the teacher list, and a list of students pops up in the student list. The user then selects a student, and all 5 musical eras populate of their entire repertoire. A musician is considered to have a piece in their repertoire when they have thoroughly learned a piece and given a performance of the work (preferably memorized for pianists).

This enables the user to quickly gain a visualization of their repertoire, seeing the big picture regarding their life's musical work and giving them perspective about new works to learn, their next recital program, goal-setting, big projects to undertake and a desire to resurrect works previously learned for future performances.

Users that would greatly benefit from such a database and visual list would be:

- Piano teachers and professors who want to see what their student's repertoire is, quickly visualizing where strengths and any "holes" and might be in that student's repertoire.
- Piano students interested in seeing their own repertoire for strengths and holes, and also to compare their repertoire with their peers (to such repertoire lists made public).
- Professional orchestras, recital and concert series, artist agencies, universities and music conservatories and even private teachers looking to recruit specific students with specializations or interesting recital programs.
- Pianists needing to submit repertoire list for auditions, applications for university/conservatory, competitions, etc. This gives them a list ready-made that doesn't need and is easily edited by simply the "add new work" drop-down feature from each era.

An important feature that I worked on but was not able to execute was the "repertoire selector" tool, another list-box that would analyze a pianist's current repertoire and randomly select suggested works to learn for their next recital program based on "holes" and/or other selected criteria. Additional details from the piano repertoire database that would help such a tool include:

- number of movements in a work
- total length of the work (in minutes)
- technical difficulty & musical "maturity"
- year composed

This feature has perhaps the best marketable implications for this program, as many pianists, teachers and musicians have difficulty selecting their next recital program. The selector would consider important factors in selecting a recital program and will help determine the length, difficulty, maturity and variety of the program. Some musicians choose to specialize in an era or even a specific composer if they desire, and can make that one of their features. Others like to play more obscure composers or obscure pieces from all composers, and this feature can dramatically enhance their selection process.

Implementation Documentation

Databases

The first thing I did was simply spend a few hours building a database of just a few of the most famous composers and their works for solo piano. I selected a dozen or so composers (my favorites), made sure that five musical eras were represented (Baroque, Classical, Romantic, "Impressionist", 20th Century/Modern), and added some details like the title of the piece, the catalog number (Op., K #, D #, BWV #, etc.), what type of piece it was (sonata, suite, etude, etc.), how long it was (in minutes), how many movements it contained and added a few more columns that other musical and pianist experts could "vote" on (technical difficulty, musical maturity).

	A	B	C	D	E	F	G	H	I	J
	Piano Repertoire Database									
	Piece ID	Composer	Era/Style	Title	Catalog	# of Movements	Total Length (minutes)	Special Catego	Tech. Difficul	Maturi
	3	Bach, J.S.	Baroque	Two-part Inventions	BWV 772-786	15	23			
	4	Bach, J.S.	Baroque	Three-part Inventions (Sinfonias)	BWV 787-801	15	28			
	5	Bach, J.S.	Baroque	Four duets	BWV 802-805	4	13			
	6	Bach, J.S.	Baroque	English Suite #1 in A major	BWV 806	8	22	Suite		
	7	Bach, J.S.	Baroque	English Suite #2 in A minor	BWV 807	6	23	Suite		
	8	Bach, J.S.	Baroque	English Suite #3 in G minor	BWV 808	6	21	Suite		
	9	Bach, J.S.	Baroque	English Suite #4 in F major	BWV 809	7	21	Suite		
	10	Bach, J.S.	Baroque	English Suite #5 in E minor	BWV 810	7	21	Suite		
	11	Bach, J.S.	Baroque	English Suite #6 in D minor	BWV 811	7	26	Suite		
	12	Bach, J.S.	Baroque	French Suite #1 in D minor	BWV 812	6	15	Suite		
	13	Bach, J.S.	Baroque	French Suite #2 in C minor	BWV 813	6	14	Suite		

What I envision is an extremely thorough database that contains more information than simply a composer and the name of the work. All of that is already available in many locations. What can make this unique is a database that adds a lot of detail to what a student and teacher can decide for what is best for them to learn next. There are some unwritten rules about the evolution of a pianist and musician and the order of repertoire they should learn.

For example, a student that has never learned a Beethoven piano sonata shouldn't begin working on a late Beethoven sonata. There is a huge level of musical and technical maturity they must have in order to give such a piece justice. In fact, Beethoven's op. 2 sonatas (certainly op. 2 #2 and #3 in A major and C major, respectively), despite being his first three sonatas, are actually quite technically difficult without necessarily having as much musical maturity as his late sonatas. So, this same student that has *never* learned a Beethoven shouldn't learn one of those sonatas, either. The op. 14 sonatas, the op. 49 sonatas and a couple others (maybe op. 2 #1, op. 10 #1 or #2, op. 31 #3 and a couple others) are great examples for "starter" Beethoven sonatas.

This type of detail will all be available on the database. The combination of the work being 1) a Beethoven work, 2) a sonata, 3) a low-scoring technically difficult and 4) a low-scoring musical maturity work will all be calculated in choosing that particular student's next repertoire. With my addendum that I'll detail later of a particular professor's studio revealing on the form what his other students are currently playing, the "piano repertoire generator" will also consider that particular student's peers. What one's peers in a studio or at the whole university/conservatory are playing is actually a fairly huge factor in pianists and other musicians in considering what pieces to work on next. People don't want to hear 5 different pianists at the same school simultaneously working on the Liszt sonata (which unfortunately happened this last year, but this was due to the 200th birthday celebration of Liszt last October).

In conclusion, the database is *far* from being complete; indeed, I just made a tiny dent in the thousands and thousands of piano works by just the more famous composers that are in existence. This is also in addition to the tens of thousands of works by obscure and current composers that are constantly being added to the website www.pianopedia.com. However, I have every intention of being able to make this database as complete as possible, by of course much more efficient means than simple manually entering in each one (as I did for the 434 works I entered).

I then created a new sheet titled "Student Database," where I just listed four piano professors from BYU and a few of their students I've gotten to know the last couple years. I study with Dr. Scott Holden, having got my BM in piano performance at UMKC's Conservatory of Music , and so he's allowed me to be in his studio as a Master's student. I know my own repertoire, so I completed most of the repertoire that's available

A flaw in the list is I don't have individual works (like etudes, very few actually learn all of Chopin's op. 10 or op. 25 etudes—it takes several months just to learn and perfect just one etude), so unless I've done an *entire* work or sonata, then it won't be listed. I'll have to look for a way to include partial lists and other things.

As a side note, I later added the "Piece ID" column (column A) in the Piano Repertoire Database as I was doing code, as I realized that when the columns were re-categorized, the repertoire would be altered based on line # and not on actual "piece ID" #.

I typed down the piece ID number in each musical era and separated each work by a comma. For testing purposes, I briefly added a mock repertoire for the other students listed.

	A	B	C	D	E	F	G	H
1	Teacher	Student	Baroque Re	Classical Re	Romantic Re	Impressionist Re	20th Century Re	New progra
2	Scott Holden	Kurt Hansen	3, 4, 8, 13, 33,	90, 98, 111, 11	155, 159, 163, 2	189, 194, 209, 219, 2	283, 290	
3	Scott Holden	Randy Peck	7, 23, 78	95, 107	138, 160	217	284, 293, 296	
4	Scott Holden	Yurika Ohki Taura	5, 7, 9, 55, 56,	89, 91, 102, 10	140, 248, 249	207, 212, 218	285, 300, 304, 309, 314	
5	Scott Holden	Camille Balleza	13, 52, 55	88	141	208	286	
6	Scott Holden	Halie Augustus	54, 59	89	142	209	287	
7	Scott Holden	Jackie Bodily	12, 50, 51, 59,	90	143	210	288	
8	Scott Holden	Penny Warwick	57	91	144	211	289	
9	Irene Peery-Fox	Conlan Miller	11, 51, 55, 59	351, 363, 369	241, 255, 261	324, 327, 335	301, 311, 315	
10	Irene Peery-Fox	Ben Walley	4	92	145	212	290	
11	Irene Peery-Fox	Heidi Astle	5	93	146	325	291	
12	Irene Peery-Fox	Arianne Sam	6	94	223	326	292	
13	Irene Peery-Fox	Jonathan Keith	7	95	224	328	293	
14	Irene Peery-Fox	Jared Pierce	8	96	225	329	294	
15	Jeff Shumway	Desiree Gonzalez	9	97	226	213	295	

I also added the column "New program" to feature what they are currently working on, a direct correlation from the "repertoire generator."

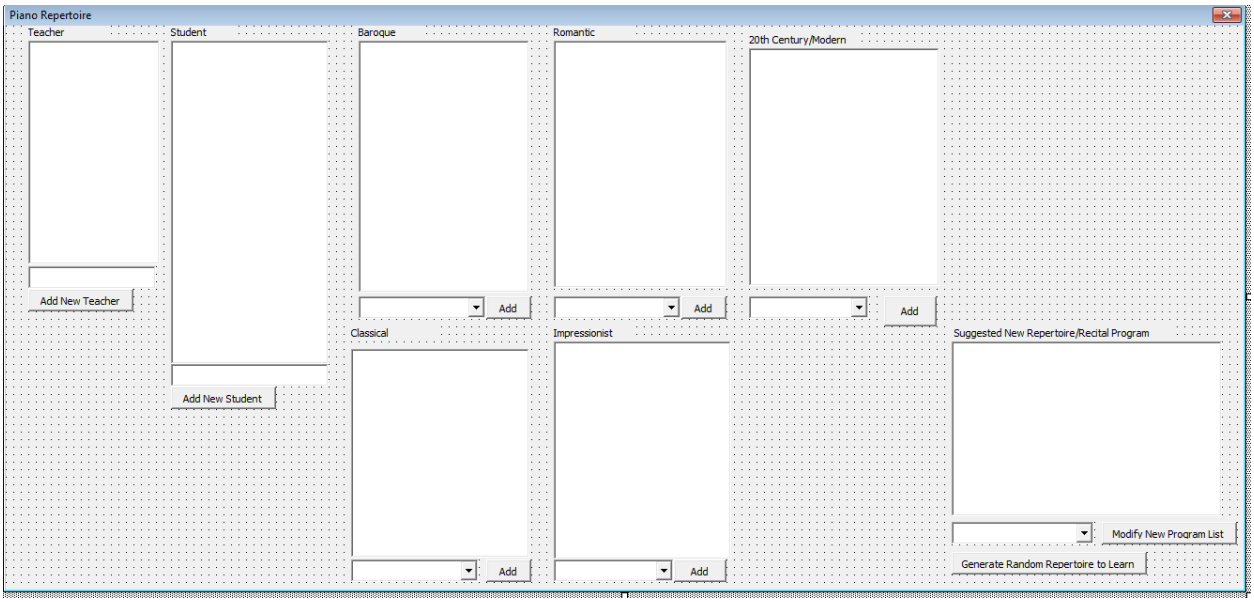
WebQuery

I added a third sheet titled "WebQuery," a simple web query as a source for users to look up works of composers on www.pianopedia.com, with links to the most famous composers of piano works linked on the sheet. Pianopedia is simply a great tool for looking up *most* (not all, but pretty dang close) works by all composers. That database, however, has gotten extremely thick and dense with extremely

obscure composers—mostly living—who think it's a marketing tool for them to show their own piano works. The problem with *Pianopedia* is anyone can add to the list.

Creating Forms

Based on learning about forms from the "UserForms" project we did for Project 4 in March, I decided creating a form was the best way to go. I created the "PianoRepertoireForm" form, which is where all the executable information is. I created a teacher list, a student list and the 5 eras that will list what a student's current repertoire is. I also added a "repertoire generator" list, but unfortunately wasn't able to populate that group because of its complexity.



Unsure of how to start this mammoth coding project, I decided to just start with the teacher list and to learn how to populate the list. I had to first figure out how to get the form to initialize, then to be able to read the teachers listed in column 'A' of the student database. Using a search method of going through all column A data, I concocted a "last row" method for a list count and simply subtracted one from the total. The teachers begin on row 2, and any duplicates needed to be prevented from appearing multiple times on the teacher list.

```
Dim lastRow As Integer
lastRow = ActiveSheet.Cells.SpecialCells(xlCellTypeLastCell).row

Dim teacherCount As Integer

For teacherCount = 2 To lastRow
    Dim teacher As String
    teacher = Cells(teacherCount, 1)

    Dim found As Boolean
    found = False

    Dim i As Integer
    For i = 0 To TeacherListBox.ListCount - 1
        If teacher = TeacherListBox.List(i) Then
            found = True
        End If
    Next i
```

An even trickier thing for me was learning how to correlate and populate the student list box based on each teacher. A student is directly tied to a teacher, and tied to that student are the five boxes of repertoire eras. A student never has two teachers simultaneously, but I would like to later add a feature of having students be independent of themselves—no teacher attached to a student. A teacher will have students attached to *them*, but not the other way around. Anyways, I had to have the teacher's list box "clicked" so that when a teacher is selected from the teacher list box, a group of students populate the student list box. A similar principle from the teachers being found was employed to the students (now from column B of the "Student Database"). I had to add the teacher value so that they were directly correlated. I also had to make sure the student list box was cleared each time.

```
studentListBox.Clear

Dim lastRow As Integer
lastRow = ActiveSheet.Cells.SpecialCells(xlCellTypeLastCell).row

Dim teacher As String
teacher = TeacherListBox.Value

Dim studentCount As Integer
For studentCount = 2 To lastRow
    Dim student As String
    student = Cells(studentCount, 2)

    Dim studentsTeacher As String
    studentsTeacher = Cells(studentCount, 1)

    If teacher = studentsTeacher Then
        studentListBox.AddItem student
    End If
Next studentCount
```

The next part was the funnest part! I had so much drive at this point, that even though it took a few hours just to figure out how to populate, I knew I was learning and was getting to the actual deliverable of "publishing" a pianist's repertoire.

I started with just trying to get the "Baroque" list to populate. To get the program to understand what the "3, 4, 8…" meant on each cell under each era, I had to create a Macro. It was a little complex, so I didn't want to figure out all the intricacies of finding all the lines of a repertoire.

```
For i = 0 To UBound(piecesList)
    Dim foundRange As Range
    Set foundRange = Columns(1).Find(What:=piecesList(i), After:=ActiveCell, LookIn:=xlFormulas, LookAt:= _
        xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, MatchCase:=False _
        , SearchFormat:=False)
```

It wasn't too difficult after finally settling with a macro, and I got a little help from a friend in the class (see below) to figure out the "split" idea. This helped populate the Baroque list with the details of the composer, the title of the piece and the catalog (in this case "BWV" for all works by J.S. Bach).

```
 Sub PopulateBaroque(studentRow As Integer)
 Dim piecesList As Variant
 Dim pieces As String
  pieces = Sheets(2).Cells(studentRow, 3)

     piecesList = Split(pieces, ", ")

     ActiveWorkbook.Sheets("Repertoire List").Select
```

…

```
Dim pieceRow As Integer
pieceRow = foundRange.row

Dim title As String
title = Cells(pieceRow, 2).Value & " - " & Cells(pieceRow, 4).Value & ", " & Cells(pieceRow, 5).Value

baroqueListBox.AddItem title
```

What really frustrated me was applying this same principle to all the other categories. I ended up trying to use the exact same method and ended up having to re-do a lot of what I had already done. Long story made short, I finally categorized the era into their own groups with private subs and had the "populateBaroque studentRow, populateClassical studentRow…" listed at the top. The rest was all gravy.

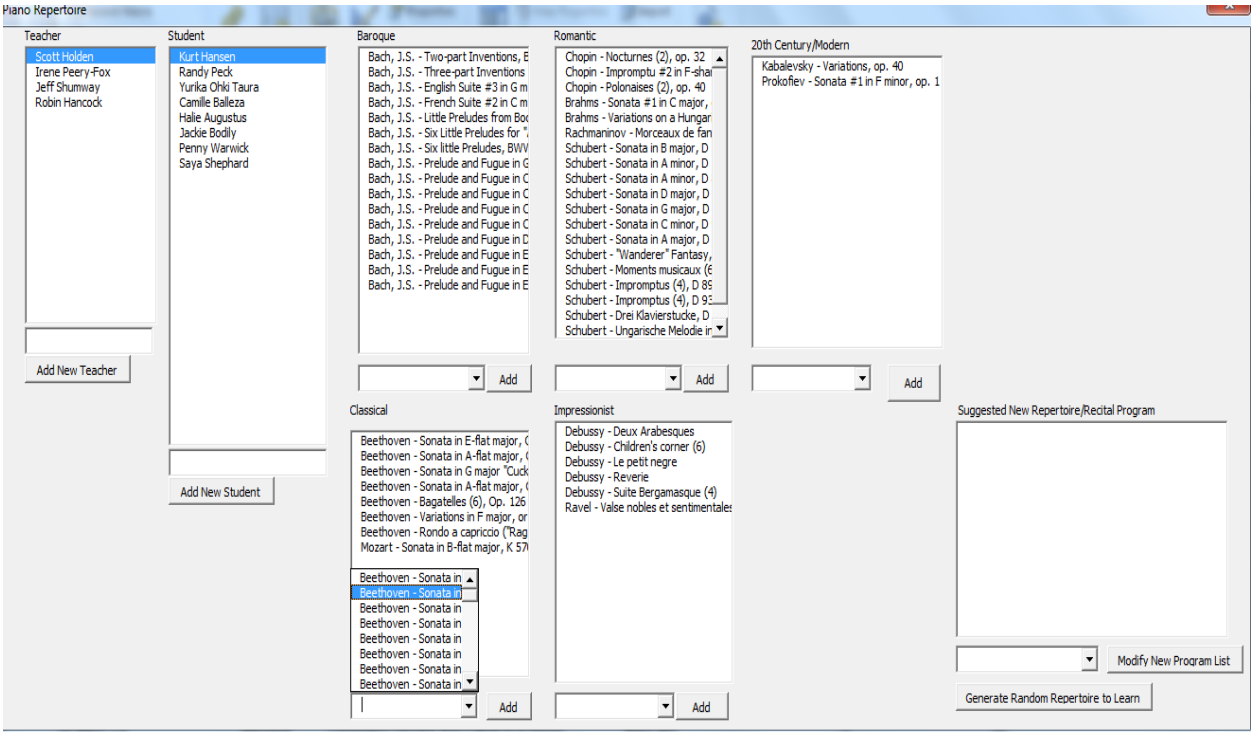
All of that took *way* longer than I anticipated, and was frustrated that I wasn't getting to the meat and best deliverable of the whole project, which was still the "repertoire predictor." I decided to create some other forms that would add and edit pianists and teachers. I also on the same "PianoRepertoireForm" decided to have the ability to add to a student's repertoire with a combo box and add feature. This wasn't entirely completed, either, but I did make some ground by using the same "last row" search method and populating the combo boxes with a range and offset method.

```
Sub populateBaroqueAddList()
Dim cell As Range
Dim sRange As Range
Dim lastRow As Integer

    lastRow = ActiveSheet.Cells.SpecialCells(xlCellTypeLastCell).row

Set sRange = Sheets("Repertoire List").Range("C1:C" & lastRow)
For Each cell In sRange
    If cell.Value = "Baroque" Then
        baroqueComboBox.AddItem cell.Offset(0, -1).Value & " - " & cell.Offset(0, 1) & ", " & cell.Offset(0, 2)
    End If
```

The final product as of now has a deliverable that looks like this:



I was able to manipulate the "add new teacher" and "add new student" command buttons to populate the teacher and student lists respectively. However, I wasn't able to populate the accompanying database on the spreadsheet when such an adjustment was made. By applying the same principles from the "UserForms" project, this wouldn't be too difficult, but I'm much more focused on getting the "Generate Random Repertoire to Learn" command button to work in the best way possible using a series of algorithms and code manipulation.

I also put some buttons on a new tab on the ribbon, but wasn't able to code it in such a way that pressing the button would cause the form to come up. Just going into the code and pressing "play" will pull up the form.

What I Learned

This project was invaluable for many reasons:

- I learned how even a seemingly simple task as populating a list-box from a database can be complex.
- I learned that in programming it's advantageous to try something very simply initially instead of trying to accomplish a difficult task all at once. Just start!
- I learned that, despite being difficult initially, once a concept is understood—it's all gravy afterwards. The hardest work comes from conceptualizing what you *truly* want to do and being able to translate that into code language. With many deliverables after that it becomes quite simple.

- I learned as I went along that there are other interfaces that might be better for what I'm trying to do than VBA, but the experience was very, very valuable in helping me understood what it's going to take to make this as thorough and yet simple as possible.

Considering the fact that I'd never had any programming experience in my entire life whatsoever before the semester, I feel very optimistic and positive about what I can do in the future with regards to programming. I believe it's something I can do, that I can actually enjoy it and that I know how to look up solutions on my own.

More future applications and possibilities to this database project are how teachers can have available on the web what all of their current students in their piano studio (or violin, or cello, or flute, or voice studio, etc.) are playing *right now*.

Other future possibilities for add-ons on this program also include aspiring composers to add their own works, similar to the *Pianopedia* model, except having a filtering process so that it won't take so long to look at such a list. Adding *many* search tools would be invaluable to such a database. *Pianopedia* isn't a very good searching tool, you just click on a letter, click on a composer and look at the huge list (not really organized other than by order of catalog #) and then click on a work to get a *little* background information.

Assistance

Riley Jenkins, friend and classmate in section 1 of MBA 614, helped me with some of the coding issues. He showed me some tools of dealing with database manipulation, initiation of the first list-box population from database and also with conceptual ideas. Riley helped with about 25% of the code.

I loved working on this project, and learned a *ton* about not just how to program, but how to think about and conceptualize methods on my own. I'm excited to continue working on this project and really want to make a website out of this idea.