Aaron Hunt

Final Project

Executive Summary

I wrote a couple of different programs for Discover Financial Services. Most popularly known for the credit card side of the business, I work in the Student Loans and Personal Loans side. I saw inefficiencies in many different areas of the business, so I decided to use some of the VBA skills that I learned to solve them.

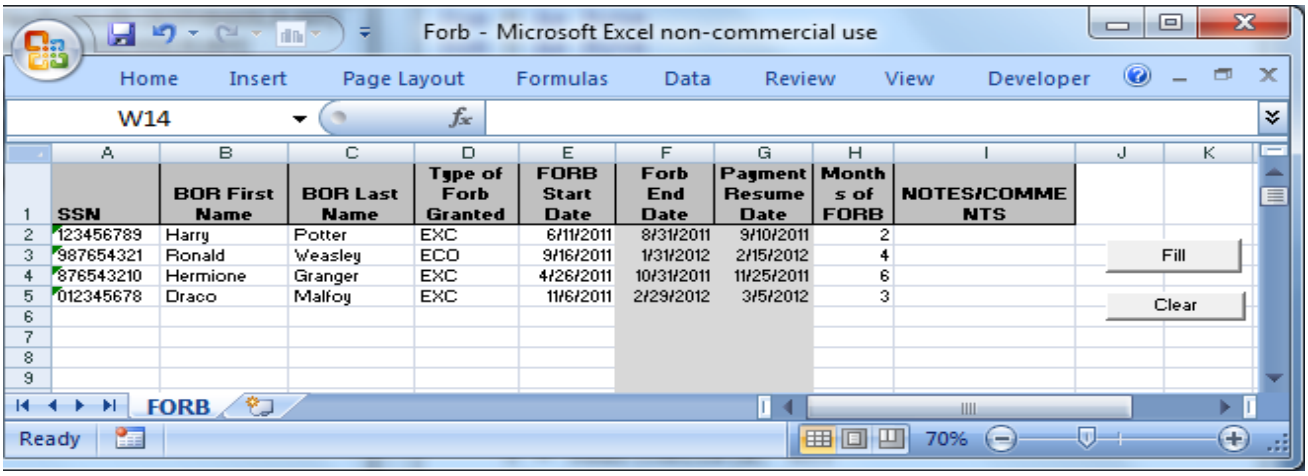
I started by creating a file that would calculate the dates needed in order to process a forbearance on the student loans. The person in charge of this was fairly slow with math, and tended to mess up when placing the dates. So I wrote a file that would find two of the dates for her, which saved close to two hours per week.

I then found that letters were being created with an excel merge file, but there were many defects found in multiple letters. I created an excel file that would take the data input and reorganize it and format it to the correct style and then merge into the word document. This decreased defects from 2% to zero.

I found that audits needed to be built for many different products. Since we already had a lot of the information being reported on a daily report I decided to use if then statements to systemically audit 85% of the process for one of the products that received a detailed report daily. I also created a report that would determine the passing rate of each account as well as the total per the team. While improving the file for the audit I decided to create a separate tab that would sort out the different accounts worked by different people to make it easier to evaluate each person.

To reduce the time wasted opening files to check on the productivity of different agents I created a vlookup that would be updated when the button was clicked, and it was able to search a file that kept track by days. This vlookup was able to find the right day every time without any work on the user's part.

Implementation Documentation

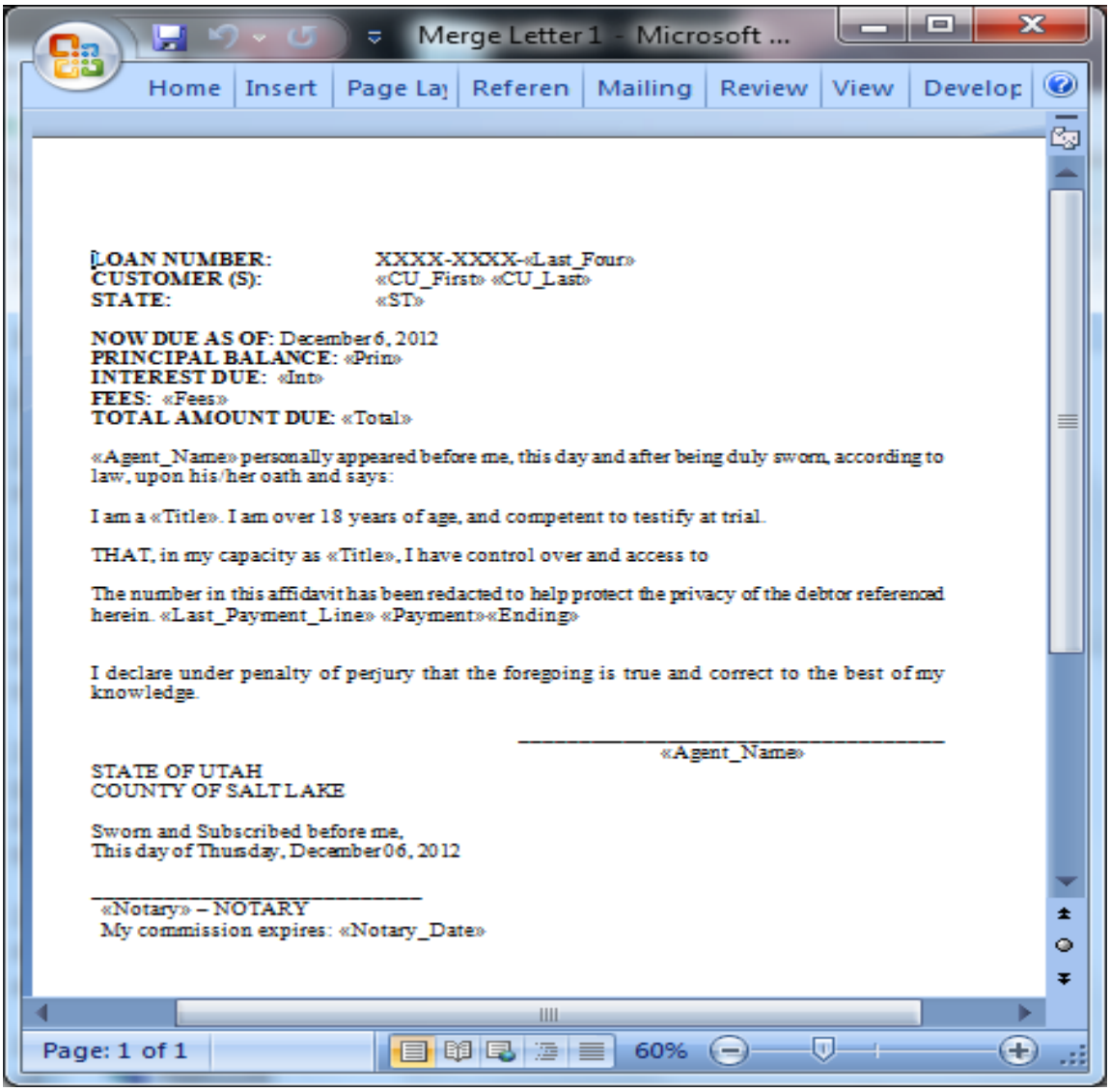


SSN	BOR First Name	BOR Last Name	Type of Forb Granted	FORB Start Date	Forb End Date	Payment Resume Date	Months of FORB	NOTES/COMMENTS
123456789	Harry	Potter	EXC	6/11/2011	8/31/2011	9/10/2011	2	
987654321	Ronald	Weasley	ECO	9/16/2011	1/31/2012	2/15/2012	4	
876543210	Hermione	Granger	EXC	4/26/2011	10/31/2011	11/25/2011	6	
012345678	Draco	Malfoy	EXC	11/6/2011	2/29/2012	3/5/2012	3	

This is a screenshot of the excel file that would calculate the dates for the user. All the user had to do was fill in the start date and how many months would be used and the other two dates were calculated for them. This was

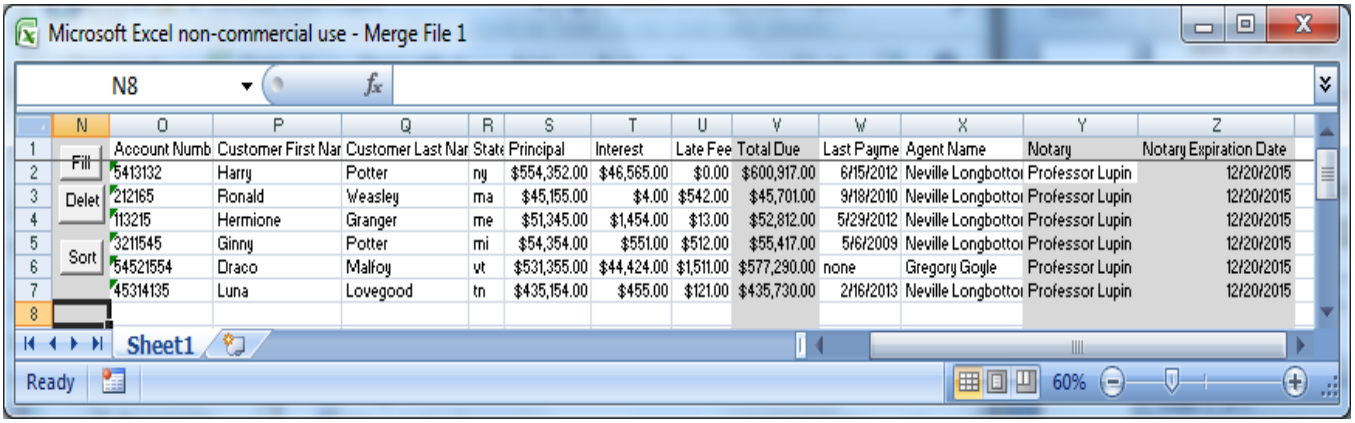
accomplished by setting up a do loop that would look to see if there was anything in column E and it would insert the dates into columns F and G until there were no longer dates. The way that it calculated the other dates was by breaking the date into months, days and years and manipulating the date accordingly. The process is very simple, by taking the start date and adding the number of months listed in column H it went to the end of that month. I even decided to throw in the fact that if it ended in February on a leap year that it would insert the 29th of February. The Payment Resume date was also simple, but required manipulating the month, day and year. It goes to the month after the forb end date and then starts the day in the month before when the start date was. I used several if then functions to make this happen.

This report is sent to a team that runs it through a system in order to process the forbearance on the loan. By having the agent simply input the start date it saves any calculation, because they are able to see the start date on their screens. This saved countless hours for the person currently working on the system, who doesn't do very well with math, but should also be helpful to anyone else that uses it.



The second part of the project was the merge files. These were giving users problems because the date would sometimes show up in the numerical count format, and the dollars wouldn't always be in the dollar format. As well, if there had never been a payment made on the account then it would insert 12:00 AM instead of saying there hadn't been a payment.

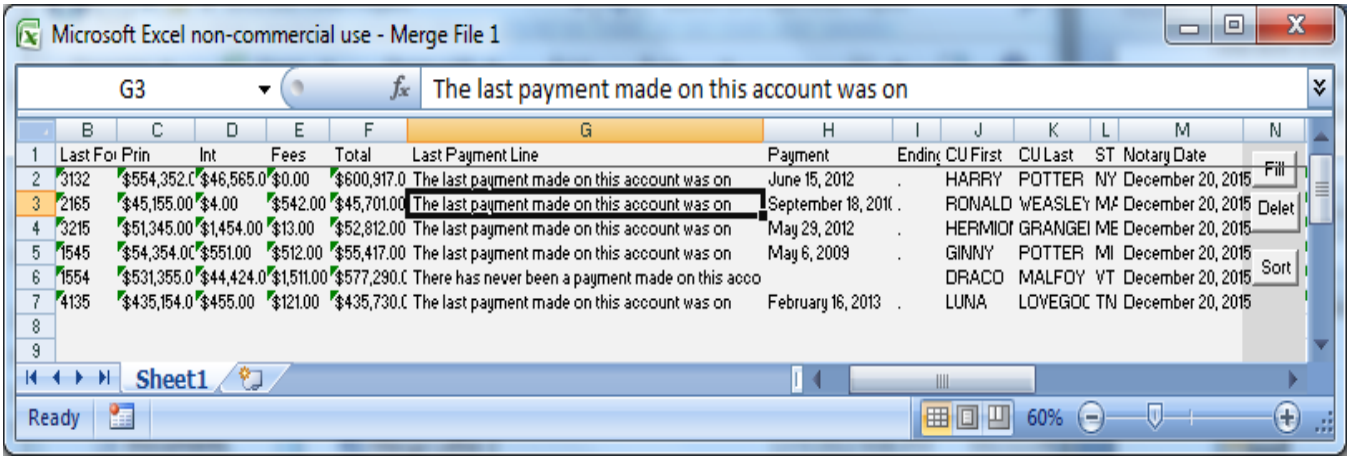
I was able to use VBA to decide that if there was never a payment then it would insert the line "There has never been a payment made on this account.", otherwise it would insert "The last payment made on this account was on " followed by the date inserted. There was also the problem that my supervisor wanted the date to read as (Month Day, Year) format. This required some If then statements in VBA as well.



	Account Numb	Customer First Nar	Customer Last Nar	State	Principal	Interest	Late Fee	Total Due	Last Payme	Agent Name	Notary	Notary Expiration Date
2	5413132	Harry	Potter	ny	$554,352.00	$46,565.00	$0.00	$600,917.00	6/15/2012	Neville Longbottoı	Professor Lupin	12/20/2015
3	212165	Ronald	Weasley	ma	$45,155.00	$4.00	$542.00	$45,701.00	9/18/2010	Neville Longbottoı	Professor Lupin	12/20/2015
4	113215	Hermione	Granger	me	$51,345.00	$1,454.00	$13.00	$52,812.00	5/29/2012	Neville Longbottoı	Professor Lupin	12/20/2015
5	3211545	Ginny	Potter	mi	$54,354.00	$551.00	$512.00	$55,417.00	5/6/2009	Neville Longbottoı	Professor Lupin	12/20/2015
6	54521554	Draco	Malfoy	vt	$531,355.00	$44,424.00	$1,511.00	$577,290.00	none	Gregory Goyle	Professor Lupin	12/20/2015
7	45314135	Luna	Lovegood	tn	$435,154.00	$455.00	$121.00	$435,730.00	2/16/2013	Neville Longbottoı	Professor Lupin	12/20/2015

This is what the user saw when putting in the information, which they were usually able to just copy and paste in sections from a report that was generated for them. Part of the process was me pulling in the Notary Expiration date based on which Notary they were going to use. In order to prevent any problems with spelling I used an array to generate the expiration date, and since I knew they would use the same Notary for the whole list I inserted the same notary name and date to prevent them from having to re-insert it.

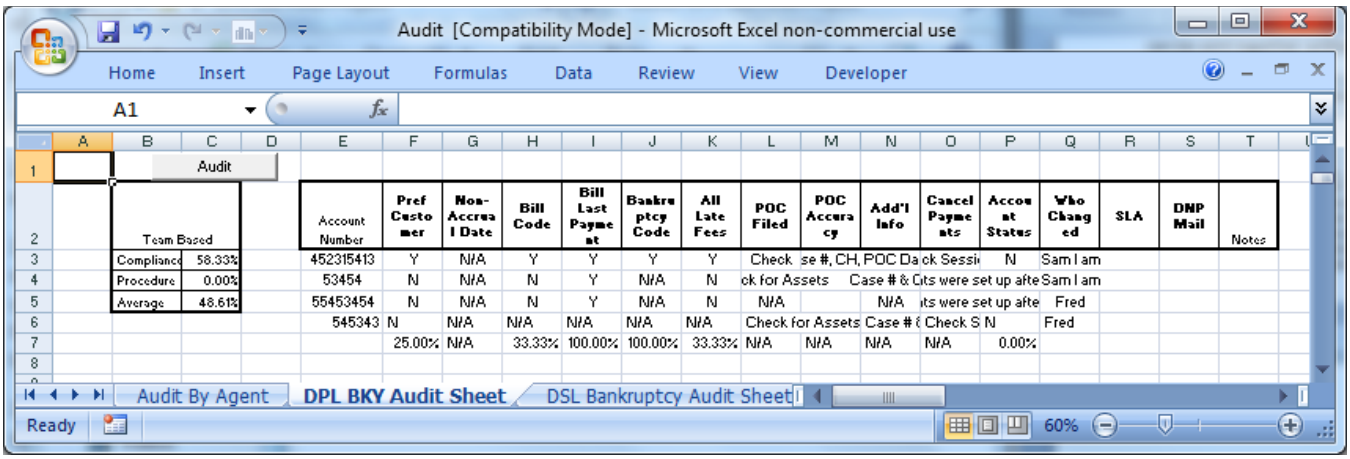
I have three buttons on there to facilitate in the process. The "Fill" button will take the inserted values and manipulate them to what needs to be inserted into the Word document. The Delete button will clear all of the data and re-insert the headings and buttons. The sort button is used to put all of the states in order so that they can be sent together without having to sort through all of them.



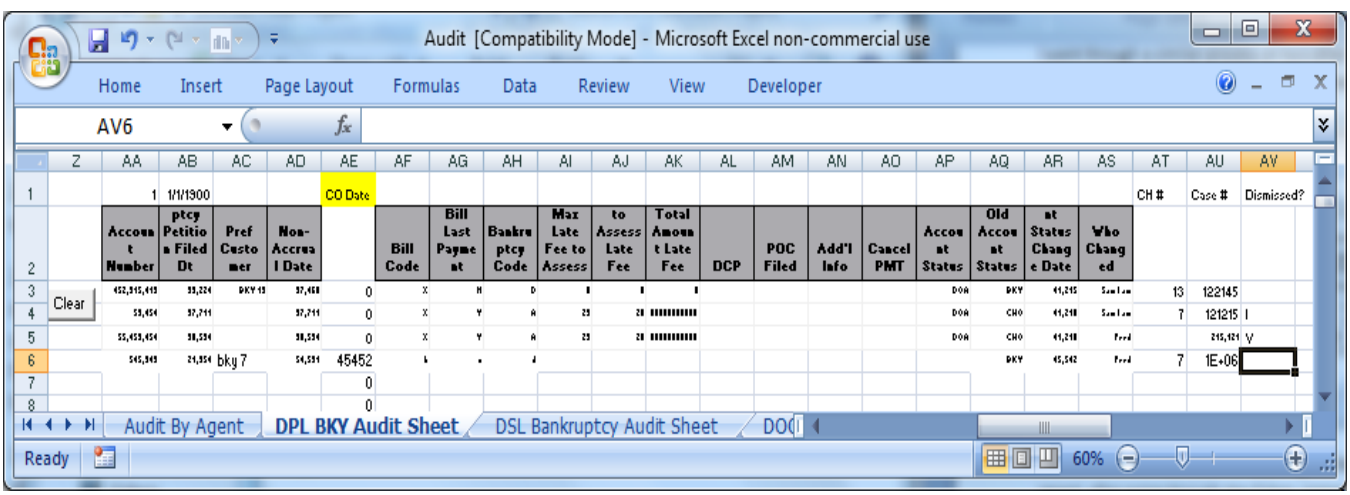
	Last Fo	Prin	Int	Fees	Total	Last Payment Line	Payment	Ending	CU First	CU Last	ST	Notary Date
2	3132	$554,352.0	$46,565.0	$0.00	$600,917.0	The last payment made on this account was on	June 15, 2012	.	HARRY	POTTER	NY	December 20, 2015
3	2165	$45,155.00	$4.00	$542.00	$45,701.00	The last payment made on this account was on	September 18, 201	.	RONALD	WEASLEY	MA	December 20, 2015
4	3215	$51,345.00	$1,454.00	$13.00	$52,812.00	The last payment made on this account was on	May 29, 2012	.	HERMIOI	GRANGEI	ME	December 20, 2015
5	1545	$54,354.0	$551.00	$512.00	$55,417.00	The last payment made on this account was on	May 6, 2009	.	GINNY	POTTER	MI	December 20, 2015
6	1554	$531,355.0	$44,424.0	$1,511.00	$577,290.0	There has never been a payment made on this acco			DRACO	MALFOY	VT	December 20, 2015
7	4135	$435,154.0	$455.00	$121.00	$435,730.0	The last payment made on this account was on	February 16, 2013	.	LUNA	LOVEGOC	TN	December 20, 2015

This is a screenshot of the information that was generated for the user. This process used to take a long time to do, and we would do it about six or seven times a month. By fixing the formatting and inserting the notary and expiration date it solved countless headaches, but also prevented a lot of defects. There would be about 2 to 4 papers out of 100 that would be printed with the wrong format or with 12:00 AM in the last payment line of the document. I was able to get around all of those problems. I also was able to make sure that the user could print all of the documents in Word and it would only print the number that was in the excel file. We never had to worry about printing a blank form.

I went through a similar process on two other merge files that did similar things but had unique features. The learned process was the same for both of them.

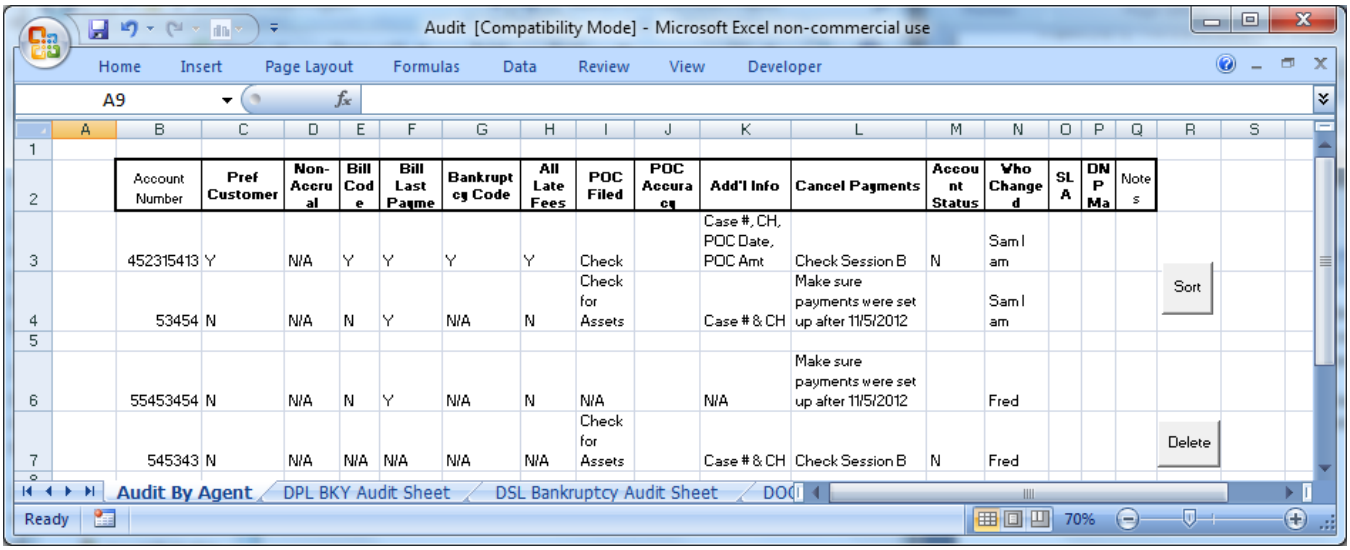


This is one of the tabs used for auditing the bankruptcy process. This section showing is generated by a system report, after going through the if then statements to decide if it was done correctly (Y), incorrectly (N) or had no impact (N/A). This is a screenshot of where the report is pulling from:

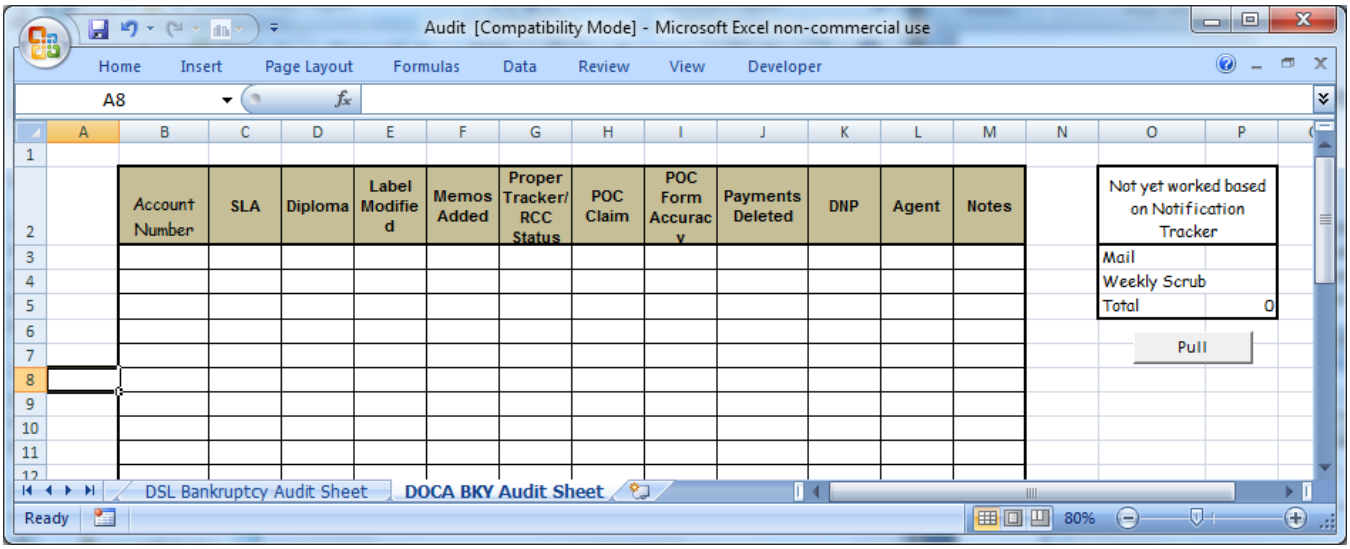


After going through the report there was VBA coding in place that generated the audit scoring formula on the bottom of the set of accounts to show what the passing rate was per criteria. Before pasting over the Y, N or N/A it would clear the section that was currently there and then fill in.

This saved countless hours that would have been spent each time there were new accounts to audit. By being able to click the clear button and have the information from the old report delete itself so the user could insert the new report and click Audit fill to do all of the work. It even told the person what the mistake was that caused a defect in the audit if they received an N. I also used VBA to put a period on the cell below all of the date columns because they had to be reformatted each time and to ensure that I got each cell in the column reformatted I would use the xlDown function. This has worked rather well and been very effective.



This is a screenshot of what we would do with the audit once we had filled in all of the information. This would take all of the accounts being audited and sort them by agent and then put a space in between each agent in order to ease the process of auditing per agent. This would pull all of the accounts from the "DPL BKY Audit Sheet" and then after placing them all on the sheet it would go through and put the space after every time a new agent was listed.



This is another one of the tabs and the unique thing to this tab is that it would find the accounts that still needed to be done in the respective tracker. By clicking the pull button it would vlookup the excel file and find that day's

date, and pull the cell values of the respective section in the work tracker. This was very useful because it saved a lot of time of opening up the documents to check throughout the day. I used VBA coding to determine the use the day and find the day of the month, then look that up in the respective work tracker. I used the concantenate functionality multiple times to ensure that the vlookup changed depending on the day, and that it would look in the correct tab of the other file depending on the month and year.

I created the same functionality of the DPL BKY Audit Sheet and the DOCA BKY Audit sheet for the Probate process of each product. Overall I used seven different tabs to audit the two separate processes for three different products.

Discussion of learning and conceptual difficulties encountered

I was able to accomplish most of this work by referring back to the different homework assignments that I had done throughout the semester. I honestly had to play with things a few times on all of the documents before I was able to get them to work. I had to ask Professor Allen how to insert a cell value, that was built from a concantenated parts and held a function, so that it would run automatically because using the ".value" just pasted it as a text format. He told me to use the ".formula" and then it worked great!

I also have to admit that there were a few mistakes in the merge files that we caught and I had to go back in and add more protection. This helped me to learn that it is necessary to try multiple scenarios to ensure that your VBA code is robust enough to handle them. But I was able to figure out the problem on my own pretty quickly as soon as I saw what the result was and what values caused it.

By having worked on the several homework assignments and projects throughout the year I was well prepared to solve all of these problems, and working through these really helped to engrain the solutions into my brain.

Assistance

Other than the help from Professor Allen on using ".formula" instead of ".value" I was able to do this entire project on my own using the classroom homework assignments and projects as guides.

Write-up Detail

As discussed earlier, in the Implementation Documentation section, I went about trying to solve multiple little problems and a few larger ones using the VBA skills that I learned this semester. I was able to generate specific dates, manipulate data to be reformatted or changed to what I needed it to say for a letter merge, do several if then statements to audit a report, use vlookup to pull from separate tabs and sections depending on the month and day, and several other smaller bonuses to improve the processes for Discover Financial Services.

I know that these systems are going to be around for a while, and that the users know how to use them properly. They have been very efficient in saving time and money for the company, and have greatly reduced errors on the part of the team. I estimate that these simple solutions have saved close to 50 hours a month of manual boring labor that is very detailed and likely to have received errors.