Cason Graff

Dr. Allen

MBA 614

Final Project: T-Mobile Bill Calculator

Executive Summary

In order to minimize the cost of cell phone service for myself and several others, I have a T-Mobile family plan that allows me to share phone service with a group of 4 other people. All of the minutes on the plan are shared. There are also some customized parts of the plan such as the data package. Some people have 3 or 4 g data packages while some only have the basic Edge service. Some people have smart phones, while others just have the old fashioned flip phone. As a result, the amount each person pays is different. There are also other charges that vary from month to month such as taxes and other fees and charges for using too much data or too many minutes. Because of this, the price each person pays fluctuates monthly depending on the circumstances. It can consequently be a pain to manually log in to T-Mobile every month, calculate the amount that each person owes, and notify each person of the amount and due date. There have been times when we've missed the due date because I never got around to figuring out the bill. My project saves time and hassle by taking care of the tedious task automatically. The program will log in to the T-Mobile account, navigate to the bill for the current month, extract the information into an excel worksheet, and create a summary page that shows the relevant information for each person on the plan. The program will then send out an email and text message to each person, notifying them of the amount they owe for the month and when it is due.

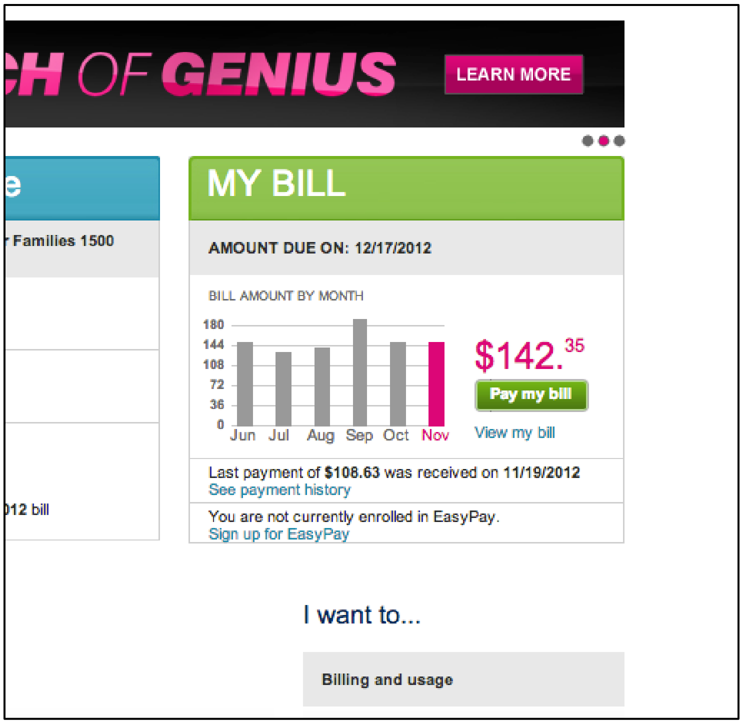
Implementation Documentation

Creating the VBA code to successfully accomplish this task consisted of three main parts:

1. Utilizing Internet Explorer to log on to the T-Mobile site and retrieve data
2. Using selected parts of the data to calculate each person's obligation and display a summary of that information.
3. Sending a customized text message and email to each person indicating the amount they owe and when it is due.

In order to provide an easy way to run the program, I used the CustomUI program to modify the ribbon, inserting a button labeled "Process Bill." The button is linked to the tmobile sub procedure which sets the process in action.

The first task was to find the data on the T-Mobile website. Using the agent toolkit and the lessons we learned in class, I created an Internet Explorer object, opened it, and navigated to T-Mobile's login site. I then performed searches of the page source code to find the username and password boxes, then had Excel input values into the boxes using the element ID's. Once logged in, I had to find various other links in the code to navigate to the page that showed a detailed view of the most recent bill. I then imported the entire page into Excel, again using the agent toolkit and the import page method. I then deleted the billdata tab to keep it clean.



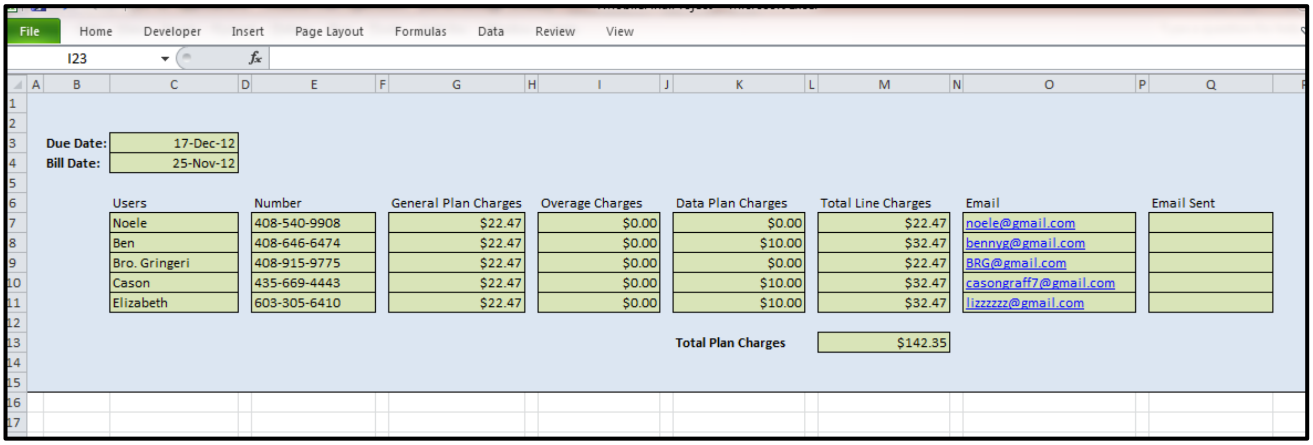
Last payment of $108.63 was received on 11/19/2012

You are not currently enrolled in EasyPay.

Figure 1 T Mobile Site

Once the data was imported onto a new worksheet, it was necessary to search through it and pull out the information that would be needed to calculate each person's individual liability. I was able to look through the data on the page and find unique values that were close to the values that I wanted to retrieve. For example, one required number was the total amount due for the month. I found that the cell just to the left of the actual total was a cell that contained the text "Grand Total." Because this was the only instance in the worksheet of that text, I was able to do a search in the VBA code for "Grand Total," then offset one cell to the right so that the total was the active cell. I then set the active cell as a variable representing the total amount due. I repeated a similar process to retrieve the due date, bill date, general plan charges, overage charges, taxes and surcharges, other charges, and discounts. Some of these values were specific to each user, in which case I created a loop to pull one person's information at a time and input it into the "Data" tab before moving on to the next person.

I then wanted to create a nice data page where I could display the pulled information and perform the calculations to arrive at the amount due for each person. I created code so that each time the code runs, new lines are inserted into the top of the data page, and formatting is applied to make the new section look good and be consistent with the data from the previous month. This also allowed me to keep an easily accessible record of the bill information from the previous months. I wrote code to place the pulled data about the phone bill into the new data section of the worksheet. All of the charges fall in to one of two categories: general or line specific. All of the general charges are lumped together then divided by the total number of people to get a general charge per person. The remaining charges, such as overage charges, are allocated to the specific line that had the overage using loops to find the values for each user individually.



Due Date:	17-Dec-12
Bill Date:	25-Nov-12

Users	Number	General Plan Charges	Overage Charges	Data Plan Charges	Total Line Charges	Email	Email Sent
Noele	408-540-9908	$22.47	$0.00	$0.00	$22.47	noele@gmail.com	
Ben	408-646-6474	$22.47	$0.00	$10.00	$32.47	bennyg@gmail.com	
Bro. Gringeri	408-915-9775	$22.47	$0.00	$0.00	$22.47	BRG@gmail.com	
Cason	435-669-4443	$22.47	$0.00	$10.00	$32.47	casongraff7@gmail.com	
Elizabeth	603-305-6410	$22.47	$0.00	$10.00	$32.47	lizzzzzz@gmail.com	
				Total Plan Charges	$142.35		

Figure 2 Data Summary Form

I created a section on the Execute worksheet where basic information about each user on the plan is stored. This information isn't directly on the bill. The bill lists only the phone number for each line. Accordingly, the code finds the phone numbers in the bill, then uses those numbers to search the Execute worksheet and return the name and email information about each person to be populated into the Data worksheet with the rest of the bill summary.

The final step was to send a customized email and text message to each person notifying them of the amount they owe, when it is due. I did this with through the use of the tools that we received in class. At the beginning of the sub, I put in a messagebox that asked the user if they desired to send notifications. This allows the user to review the information and provides a way to run the program even if you don't want to send notifications at the time. If yes is selected, a userform appears asking the user to input the gmail credentials to be used to send the message. I then created the message that I wanted to send. This was done in Notepad. Because I have a very informal relationship with each person on the plan, it was a rather simple message so that it wouldn't seem like a canned or overly formal message. I put certain points in the message where the person's unique information could be substituted out. This included the name, amount owed, and due date. VBA code was then used to loop through and replace the values in the message as each one was sent. The program sends both a text message and an email to each person.

Sending the message involved sending information to Gmail to be sent to the intended recipients. Much of this code was taken from in class exercises and then adapted for the current circumstances. A loop is used to cycle through the row numbers of the range that contains data for each person on the plan. Using the row number, the information needed to send the message is loaded into the sendGmail function, along with the login information from the gmail userform mentioned previously. The sendGmail function uses the Google SMTP server to send the email and text message. The function is designed to return a Boolean value that will indicate whether the email was successfully sent. If the sendGmail function returns an error, the function value is false and that value is sent back through the sendonemessage function to the sendMessages sub procedure which will then write “Bad Email” in the row of the message that had the error. If the message was sent successfully and no error messages were returned, the current date and time is entered on that person’s row.

Learning and Conceptual Difficulties

I was able to learn a great deal as I carried out this project. I thought it was very useful to think of a problem that I personally wanted to solve and then have to think through each of the steps to make it work. Though we talked about many of the concepts in class, this project really helped me become more comfortable with approaching problems and using what I’ve learned to create a solution.

Exploring and working with the web page code was something that was initially very fuzzy to me. I knew how to open up and access the code, and quickly learned how to find the elements I was looking for through searches, but it was a little more difficult to learn how to interact with the needed objects and even just figure out exactly which part of it I needed to use. One particularly troublesome part was learning how to deal with the incredibly slow loading T-Mobile site. For some reason or another, the waitforload sub wasn’t sufficient and the code would try to execute before the page was fully loaded and consequently return an error as it couldn’t find the required link. I solved this problem by updating the wait timer in the waitforload sub so that it waited a little longer regardless of whether it thought the page was loaded. I also used multiple waitforloads to compound the default 2 second after load wait.

Another good learning opportunity came about as I organized the data so that it would be logical and well presented. I learned to deal with the challenge that the lists of people on the Execute tab and the Data tab were in different orders. I got more efficient at using loops to find the needed row from the Execute tab, and then use that row to transfer all the needed information.

I ran into a few hiccups as I came across different formatting schemes in the data. I wanted the data to appear one way in the form so that it was more user friendly, but this caused errors such as with the phone numbers that couldn’t be turned into email addresses with the “-“ included. I used replace functions to alter the formatting to make the program work.

An additional task I would have liked to perform would be to pay the bill with the push of a button. I feel like I could have accomplished this if I had more time to work on the project. I also would have liked to put in a system so that if the program had already been run for the current month, it would skip the data retrieval step and simply ask if you wanted to send notifications. That would be useful in case people hadn’t paid their portion yet and I wanted to nudge them to get on the ball. The last thing I would include would be a way to run the program again for any given month by selecting the desired month from a userform.

Assistance

As I was using the VBA code to navigate in Internet Explorer, I had a lot of trouble figuring out how to click the last required link which took me to the bill page that had all of the relevant billing data. The link used javascript and I was unfamiliar with how to go about interacting with the link. The previous links had all had ID's onto which I could attach and execute the click method. After trying to click through the link in various ways and searching for ways to interact with java on the internet, I went to Dr. Allen's office for help. He was able to find the right portion of the code and use execscript correctly. He was also able to make VBA attach to the new window because the link opened in a new tab.

The other form of help I received was through the class files. I was able to download and review those files which helped me understand how to perform certain functions. I could then modify the code to fit in the circumstances under which I was working. That is the extent of substantial assistance that I received with this project.

Conclusion

Overall I felt like doing this project was a good chance to apply all the things I've learned throughout the semester to accomplish something that is meaningful to me. It helped me gain confidence that I will be able to use VBA and apply it to projects that I come across in the workplace. It solidified my feeling that this class was extremely worthwhile for my career development.