Simple Budget Software

Background

This program is designed to help college students or young families manage and track their finances overtime. The problem that many college students face is having time to manage and track their money and many are living paycheck to paycheck without much thought of saving. Many students, especially at Brigham Young University, have jobs and are earning enough money but don't realize where there are spending all of their money and rarely spend the time or effort to manage and save for big purchases or emergencies.

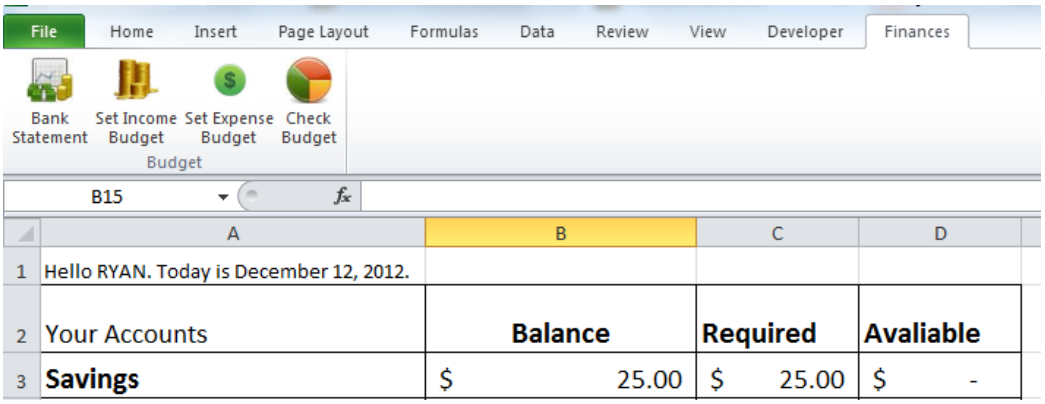
Solution Overview

This program is designed to minimize the time spent having to manage finances while still allowing the user to know where money is spent and to track savings for future financial goals. This is accomplished by pulling balances and transaction history from a bank website and allowing the user to keep track of balance history overtime. The program also uses simple user forms to allow the user to enter in a monthly budget of income and expenses which can then be used to compare with actual numbers at the end of the month using generated charts.

This program is useful because it allows the user to pull all of their financial information into one worksheet without having to log onto an online bank site and manually go through the information. The program allows the user to see visually where/what their money is being spent (on) each month in order to make improvements. It is also very useful to track account balances to avoid going under minimum balances or to track savings over time in order to make large purchases such as a house or car, or to pay off debt.

Implementation Document

The first step I performed was an excel ribbon modification in order to include a “Finances” tab which houses the four buttons shown below.



These buttons are used to automate the process of tracking finances, setting monthly budgets, and tracking actual money spent over time. The following sections will explain each button and the macros they run in detail.

Bank Statement

This button is used to pull current balances from a savings and checking account and records these amounts at most daily in a history chart in order to track savings and or account balances over time. Calling this macro with the bank statement button first brings up a form which the user can use to enter their bank account number and password in order to avoid writing these values into the code for security reasons. The macro then uses an agent in order to open up an internet explorer browser and navigates to the bank’s home page. The account number and password entered in by the user are then placed in the appropriate boxes and the agent logs into the customer account.

Once logged in, the agent pulls the user name and the date and places them in the excel work book as well as the account balances from the users savings and checking accounts. It then navigates to pages with transaction history for both of these accounts and uses a web query to bring in data from these pages and creates new sheets in the workbook to hold this information which can then be easily seen and analyzed in excel s and/or used to enter in actual numbers in the monthly budget which will be created shortly.

If the sheets for the savings and checking accounts have already been created these sheets are deleted and new ones are added holding the current information. If this is the first time during the day the user has pulled these account balances a new line is added on the “Balance” page under Balance History and the date and balance information are entered in to provide a history of savings and checking balances with the dates in a simple to track and analyze form (see below).

Hello RYAN. Today is December 12, 2012.

Your Accounts	Balance	Required	Avaliable
Savings	$ 25.00	$ 25.00	$ -
Checking	$ 963.88	$ 100.00	$ 863.88

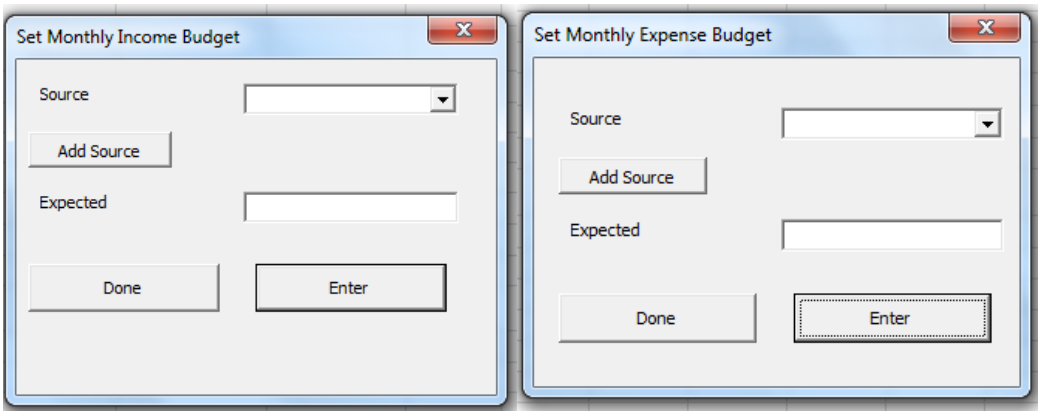
Balance History

Date	Savings Balance	Checking Balance
12/12/2012	$25.00	$963.88
12/11/2012	$25.00	$963.88

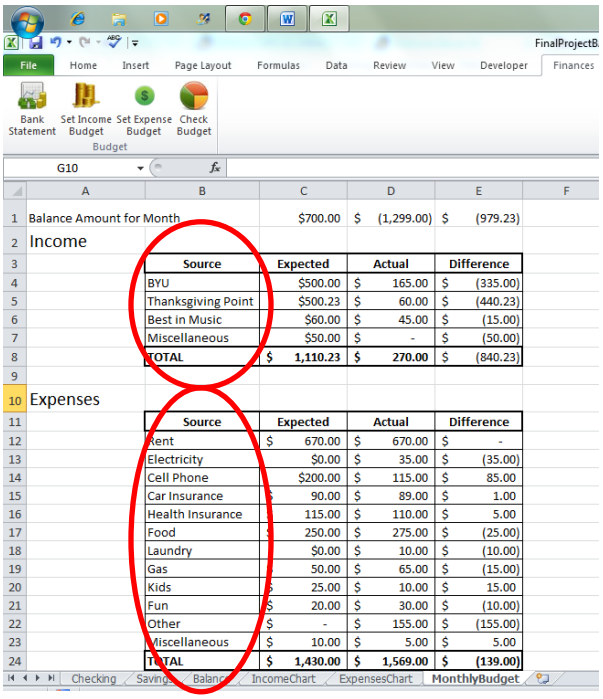
Finally the macro searches the checking and savings sheets which were entered into the workbook for any instance of the users account number and replaces it with "*******" and then erases the values entered into the accountNumber and password variables in the macro and clears the user form used to input these values.

Set Income Budget and Set Expense Budget

The function of these two buttons is essentially the same so they will be discussed together. After pressing these buttons a user form is brought up and the macro activates the "MonthlyBudget" page of the worksheet so that the user can see the changes they are making to the budget. The user forms are shown below.



The combo box to the right of the Source lable will automatically populate with the current sources listed from the Income and Expenses areas of the "MonthlyBudget" sheet shown below.



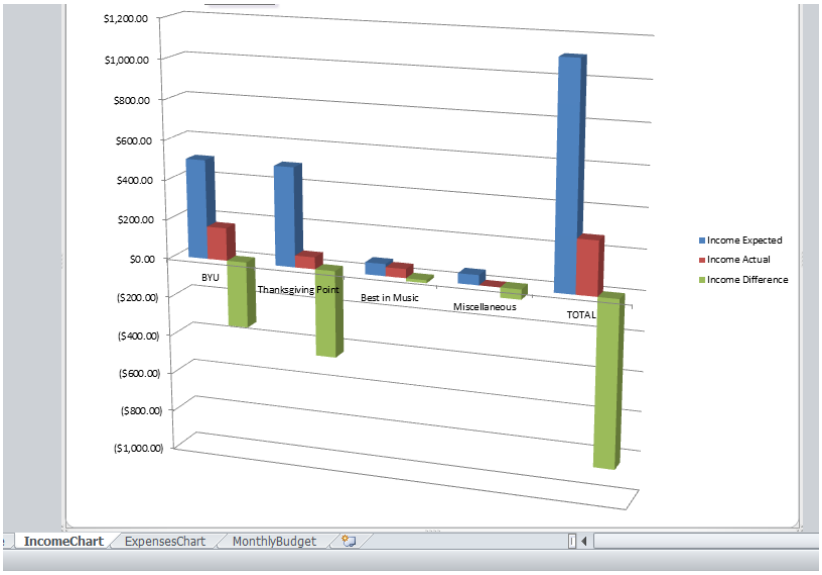
	A	B	C	D	E	F
1	Balance Amount for Month		$700.00	$ (1,299.00)	$ (979.23)	
2	Income					
3		Source	Expected	Actual	Difference	
4		BYU	$500.00	$ 165.00	$ (335.00)	
5		Thanksgiving Point	$500.23	$ 60.00	$ (440.23)	
6		Best in Music	$60.00	$ 45.00	$ (15.00)	
7		Miscellaneous	$50.00	$ -	$ (50.00)	
8		TOTAL	$ 1,110.23	$ 270.00	$ (840.23)	
9						
10	Expenses					
11		Source	Expected	Actual	Difference	
12		Rent	$ 670.00	$ 670.00	$ -	
13		Electricity	$0.00	$ 35.00	$ (35.00)	
14		Cell Phone	$200.00	$ 115.00	$ 85.00	
15		Car Insurance	90.00	$ 89.00	$ 1.00	
16		Health Insurance	115.00	$ 110.00	$ 5.00	
17		Food	250.00	$ 275.00	$ (25.00)	
18		Laundry	$0.00	$ 10.00	$ (10.00)	
19		Gas	50.00	$ 65.00	$ (15.00)	
20		Kids	25.00	$ 10.00	$ 15.00	
21		Fun	20.00	$ 30.00	$ (10.00)	
22		Other	$ -	$ 155.00	$ (155.00)	
23		Miscellaneous	$ 10.00	$ 5.00	$ 5.00	
24		TOTAL	$ 1,430.00	$ 1,569.00	$ (139.00)	

The user selects one of these sources and then inputs a numerical value in the box to the right of the Expected lable and the table in excel is populated when the user clicks enter. Clicking done will hide the user form and allow the user to make ajustments manually to this sheet or enter in actual numbers using data in the bank statement transaction sheets (checking and savings sheets). Clicking on the add source button will allow the user to type in a new source in the box to the right of the source lable and will insert a new row with this new source name directly above the Micellaneous row in either the Income or Expenses sections respectivly. The user can then enter in an expected amount for this new source exatly as the others.

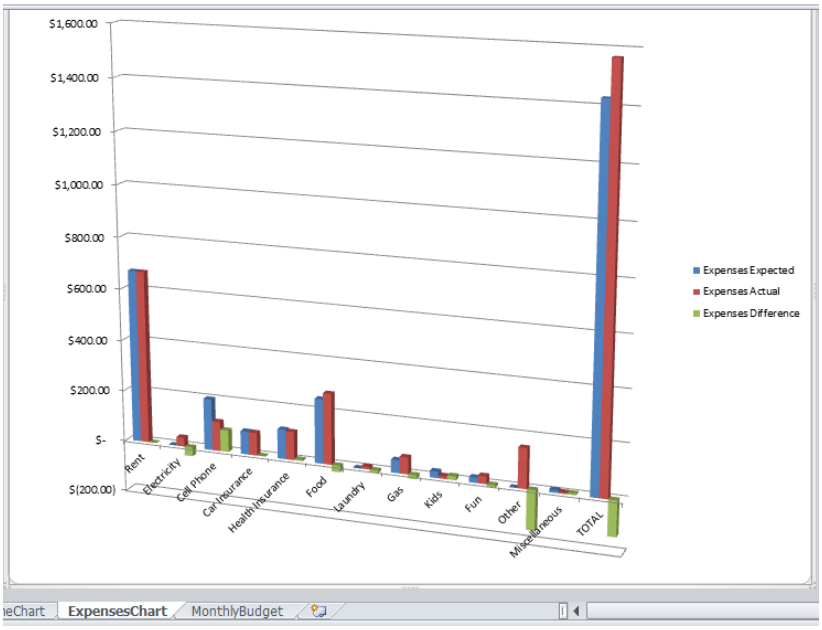
Check Budget

The user implimentation of this button is extremely simple. Once clicked the user will be provided with two bar charts showing the expected, actual, and difference between these two numbers for both the income and expenses entered into the “MonthlyBudget” sheet. This macro creates two new sheets in the worksheet to display these charts and allows the user to easily see where they are receiving/spending their money and where they have large discrepancies in their budget vs. actual numbers (see examples of charts below).

INCOME



EXPENSES



Learning and Conceptual Difficulties Encountered

This project really opened my mind to the power of VBA and the many possible applications not just for work and business but for everyday things in life. The web agent I worked with is especially powerful but also posed the most difficulty as I worked to understand how it worked and how to utilize it to navigate to different pages which were password protected and pull the relevant data either from the web page source document or using a web query built into the agent.

I was hoping to be able to use the transaction data that I got from the bank to populate the actual column on the "MonthlyBudget" sheet so that this part could also be automated but I couldn't figure out a way to classify each transaction to the different sources of income or expense without the user entering a lot of additional information so that portion will continue to remain manual until I can figure out a good way of either entering in these transactions with a source or some other way of classifying them.

This project also pulled together many different aspects of VBA which we have discussed over the semester and really solidified them so that I feel comfortable going forward and using VBA to solve problems and automate tasks to save time and increase productivity both at work and it my personal life.

Finally, this project taught me a lot about error checking and patience. Especially in using the agent and in trying to figure out how to get started on many of the aspects of my project there were many errors and frustrations as I tried many different ways to reach my desired outcome. Throughout the struggles I was able to stay focused and put in the time necessary to learn (many times through trial and error) how VBA functions and what is required to accomplish certain objectives.

Assistance

I received no assistance on the writing of this project. The agent used was written previously by Dr. Gove Allen and provided to us to use as we wanted and I found one line of code online on how to populate a combo box. Other than those two items all code was written by me, Ryan Oliphant, with some reference to previous projects we have completed throughout the semester.