

## **Executive Summary – Receipt Summarizer and Expense Report**

The purpose of this program is to provide a way to analyze my spending habits over a given month of time. One of the needs that I have is a way to categorize my spending to make my monthly budgeting process more meaningful and give an accurate estimate of my expenses. Prior to having this program, I never really knew how much money I was spending on things like eating-out, entertainment, and groceries. In essence, this program is a homemade version of Quicken and will allow me to track my purchases and evaluate my spending habits.

Here is a brief overview of the program.

The system allows the user to enter receipts on a line item basis and will summarize the entire receipt according to the receipt number, merchant used and receipt total. The system includes a return feature to accommodate any purchases that are returned to the merchant. Additionally, the user is able to create charts. This provides a visual summary of the expenditures by item category (i.e. food, clothing, eating-out, etc). Finally, the program allows the user to access the transactions that have been posted to the user's bank account (credit card, debit card, checks, etc.) and download those transactions to the worksheet where the user will be able to reconcile the bank account to their own receipt records. Ultimately this will allow the user to evaluate their spending habits and adjust their monthly budget accordingly.

### Major Features

- Enter Receipts
- Return Items
- Create Summary Charts
- Download transaction summaries from a secure bank website.

## Implementation Documentation

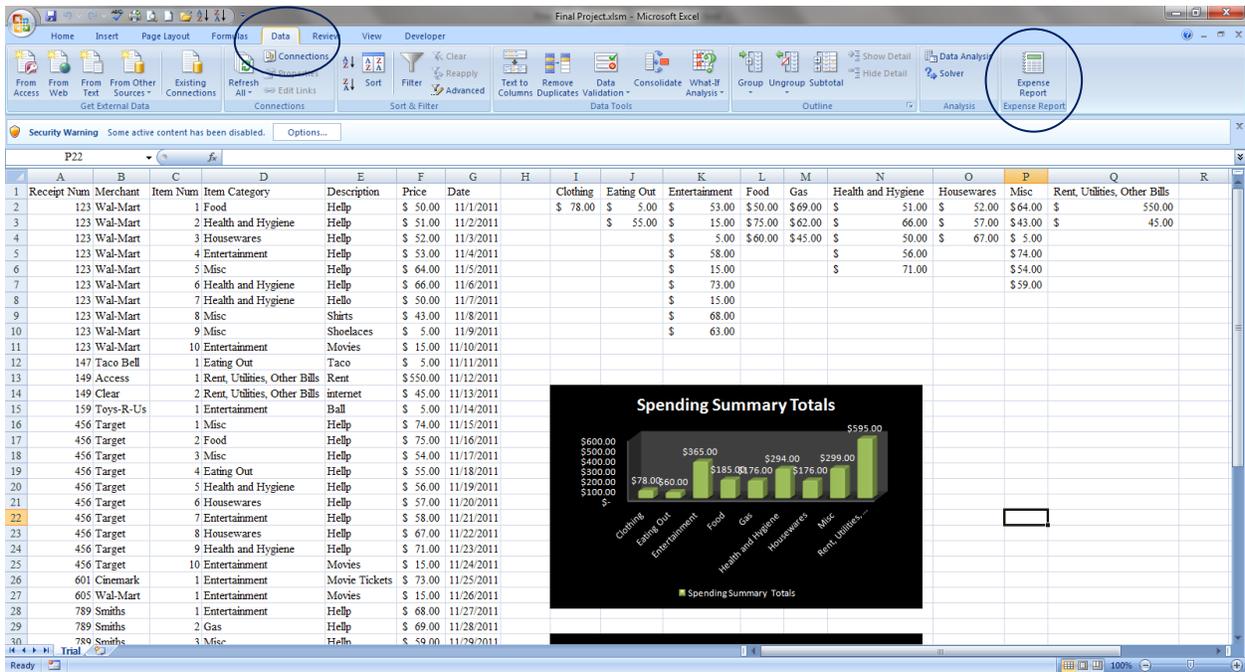
### Overview

The purposes of the features in this project are to provide a summary for all expenditures that occur in one month and to make it easy to evaluate my purchasing habits. Specifically, I incorporated a graph functionality to provide a visual summary of my spending. I also created a feature that will pull my transactions from my bank's website. This feature will allow me to verify my purchases during the month and reconcile my bank account.

The following pages describe in detail each feature of the program, how to use it, and why it was included. Below is a brief summary of the different features I have included.

- Access The program
  - Data Tab > Expense Report
- Welcome Screen
  - Features
    - Start a new Month
    - Enter new receipts/receipt items
    - Create charts
    - Download transactions from bank website
    - View receipt details
    - Return/Delete items from receipt
- Enter new Receipt Screen
  - Features
    - Enter receipt details
      - Date Purchased
      - Receipt Number
      - Merchant Name
      - Item Price
      - Item Description
      - Item Category
    - Delete items from receipts to correct errors

1. Opening the program –
  - a. To open the program the user needs to first open the excel file, and enable macros.
  - b. Navigate to the data tab and click the Expense Report Icon



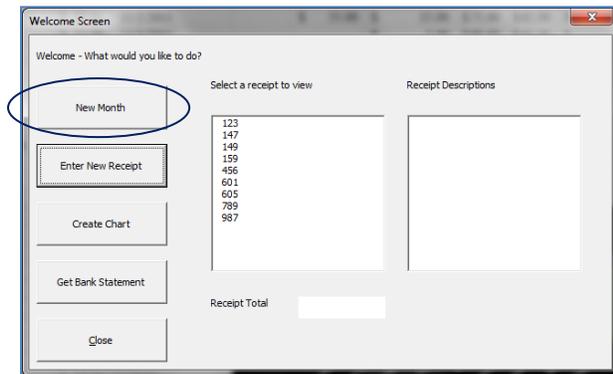
2. Welcome Screen
  - a. When the program opens, the user is presented with a welcome screen that will allow the user to access the various features of the program. I used forms to organize the program into a consolidated, user-friendly format.

The "Welcome Screen" dialog box contains the following elements:

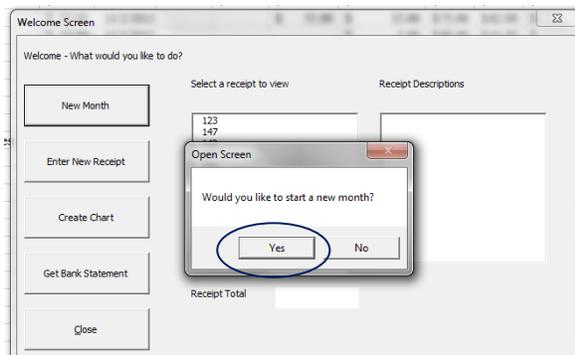
- Buttons: New Month, Enter New Receipt, Create Chart, Get Bank Statement, Close.
- Text: "Welcome - What would you like to do?"
- Section: "Select a receipt to view" with a list of receipt numbers: 123, 147, 149, 159, 456, 601, 605, 789, 987.
- Section: "Receipt Descriptions" (empty).
- Text: "Receipt Total" followed by an input field.

3. Start a new month

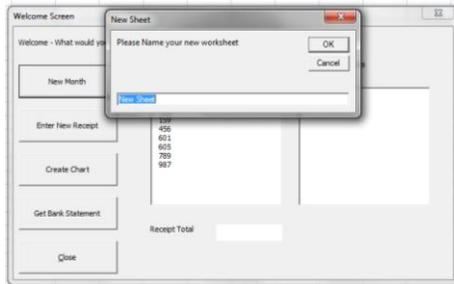
- a. If this is the first time the user will be using the program, or if the user is inputting receipt information for a different month, the user will need to access the New Month Feature.
- b. The first step is to click on the New Month button.



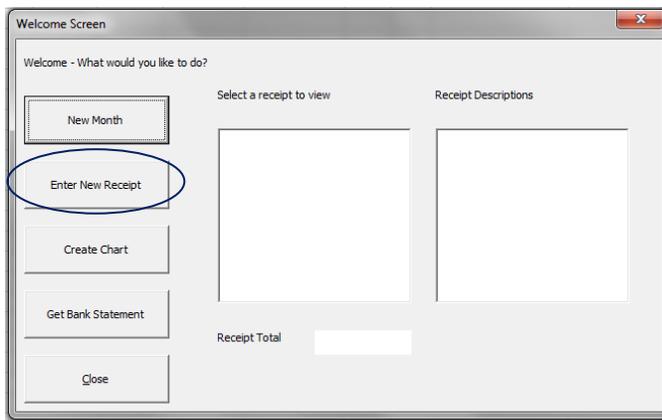
- c. Immediately a message box is presented which asks the user to confirm that the user wishes to start a new month. While this may seem a little redundant, I included this extra step account for inadvertent button pushes. I know I push the wrong button often enough and this allows the user to go back without creating a new month and the accompanying blank worksheet.



- d. If the user confirms by pressing yes, an input box pops up giving the user the opportunity to name the new worksheet. I advise the user to use the name of the month and year for which the expenses will be entered. This is not required but will help with organization.
- e. Clicking No will allow the user to stay on the current spreadsheet and work from there. This is useful if the user had more receipts for the previous or current month to enter.



4. Enter New Receipt info – Note this feature is always available from the welcome screen regardless of if the user is working on a new or previous month
  - a. To access this feature the user needs to press the Enter New Receipt button from the Welcome Screen



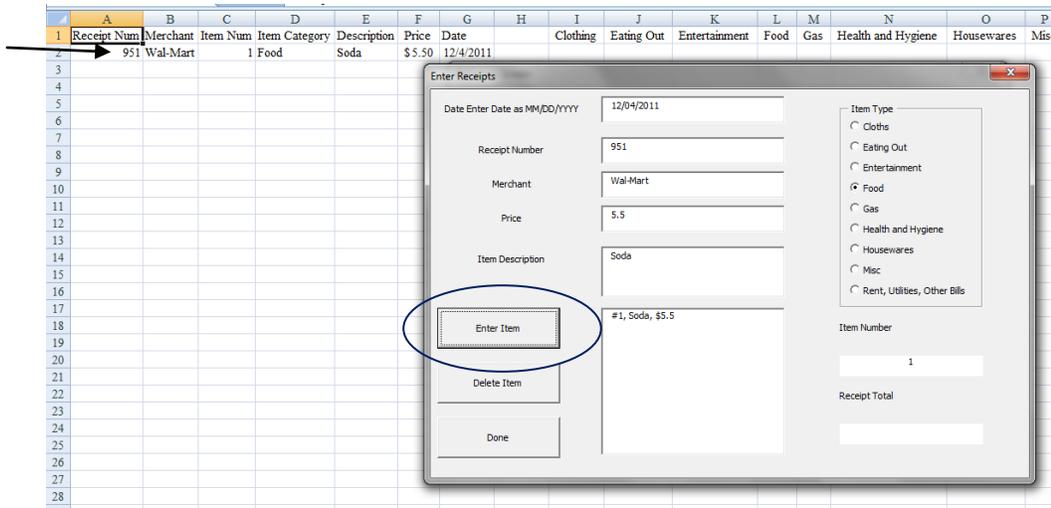
- b. This is the main feature of the program. This feature allows the user to enter receipt information onto the spreadsheet for future summarization and evaluation. This is for line item entry.
    - c. When the user presses the Enter New Receipt button, a new form opens which has fields for entering critical receipt information.
    - d. The program requires the user to enter all fields.
      - i. If any field is blank the program will prompt the user to complete the form and will not enter the information until the user completes the form.
    - e. User Fields
      - i. The main fields are:
        1. Date the transaction took place – The user should enter the date in the following format MM/DD/YYYY if the user does not enter a valid date the program will send an error to the user and request that they input a valid Date. The program will accept any valid date in any format but for simplicity and organization, the user should use the prompted date format.

2. Receipt number (this can be the number on the merchant receipt or any number the user wants, its purpose is strictly for organization in the spreadsheet).
3. Merchant Name – Will allow the user and the program to reconcile purchases with the bank transactions
4. Price
5. Item Description
6. Item Type – This allows the user to track expenses categorically and is used in the Chart Feature to show trends in expenditures.
7. Item Number – This DOES NOT represent the number of the specific items purchased. This represents the order the item appears on the receipt as it is input into the system. The only purpose of this number is to help organize the spreadsheet and as a convenience to help with item returns/deletions which is discussed later.

The screenshot shows a software window titled "Enter Receipts". It features a form with the following elements:

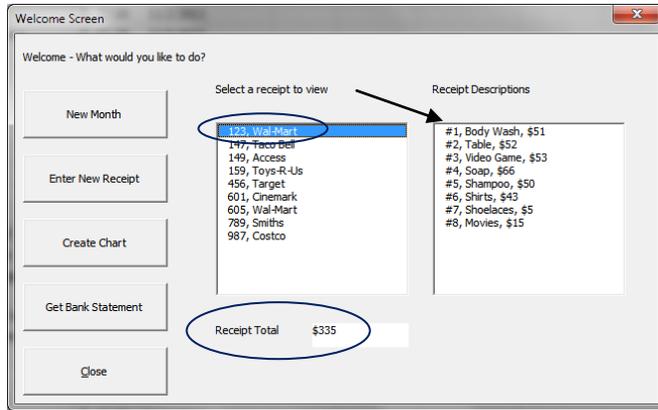
- Date Enter Date as MM/DD/YYYY**: A text input field.
- Receipt Number**: A text input field.
- Merchant**: A text input field.
- Price**: A text input field.
- Item Description**: A large text area for describing the purchase.
- Item Type**: A vertical list of radio buttons for categorization: "Cloths", "Eating Out", "Entertainment", "Food", "Gas", "Health and Hygiene", "Housewares", "Misc", and "Rent, Utilities, Other Bills".
- Item Number**: A text input field.
- Receipt Total**: A text input field.
- Buttons**: Three buttons are located on the left side: "Enter Item", "Return/Delete Item", and "Done".

- f. After filling out the form press the Enter Item button to enter the item into the spreadsheet.
  - i. In addition to being put on the spreadsheet the item now appears in the list box. This box will update for each item entered on the SAME receipt and will reset for new receipts
- g. Below is an example of entering an item into the form



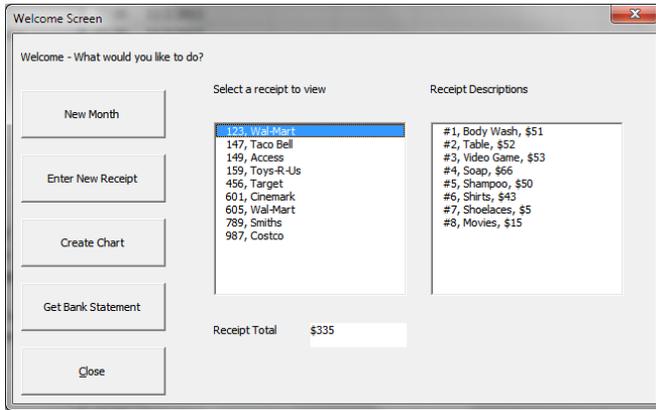
- h. Dealing with entry mistakes –
  - i. If you enter an item incorrectly or with an error in it simply select the item from the list box, press the return/delete item button, and re-enter the item.
  - i. To enter a different receipt simply change the Receipt number to reflect the new receipt
5. When all receipts have been entered, Press the Done button to return to the welcome screen.
  - i. Note that after you press done the program updates and sorts the welcome screen list box and the worksheet to reflect the changes.
6. Review Receipt Information
  - a. On the welcome screen two list boxes provide a summary of the receipts that have been entered into the program. (Note that if this is a new month there will be no items in either box). The purpose of this feature is to provide a quick summary of the different receipts that have been entered for the month. The first list box provides the receipt number along with the name of the merchant.
  - b. To view the specific items for a given receipt simply click ONCE on the desired receipt in the first list box. Doing so will populate the Receipt Descriptions box with the various receipt descriptions
    - i. The Description consists of the item description and the price. (You will also notice a number in front of the description. This is the item number and is there as a placeholder to help with item returns and is the same number as mentioned in the previous section.
  - c. The Receipt Total is displayed below the list box

Ron Barwick  
MBA 614

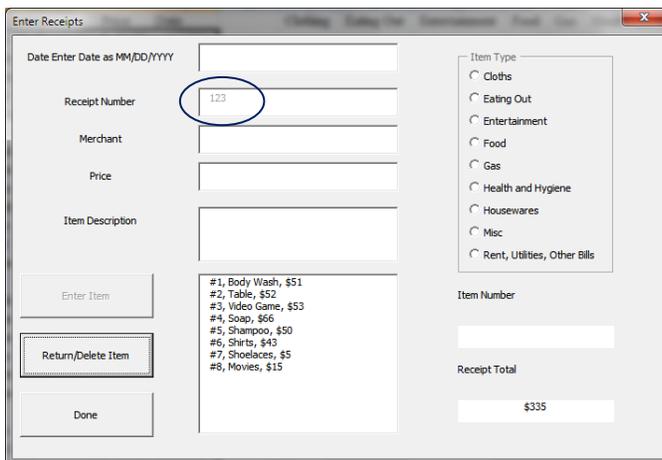


7. Return Items from a receipt

- a. Another key feature is the ability to account for items returned to the merchant.
- b. To return an item **DOUBLE** click on the item number in the 1<sup>st</sup> list box (the one with the receipt number and merchant name)



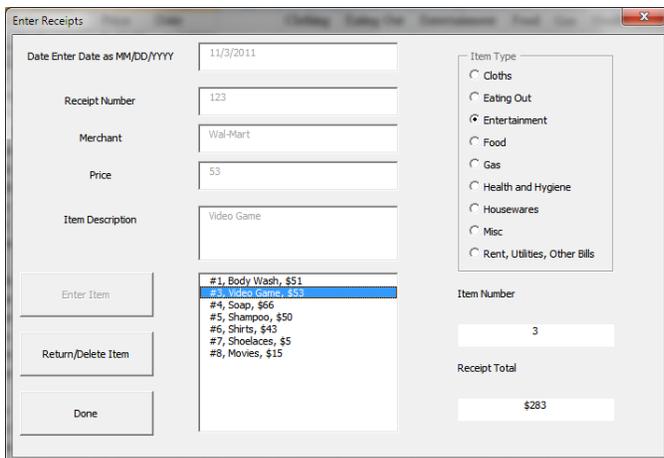
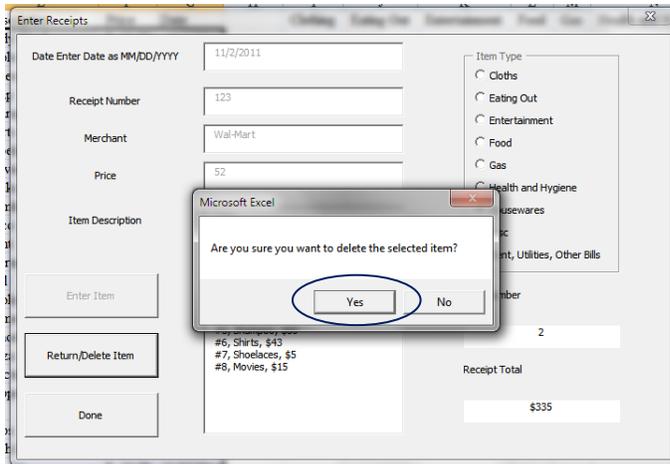
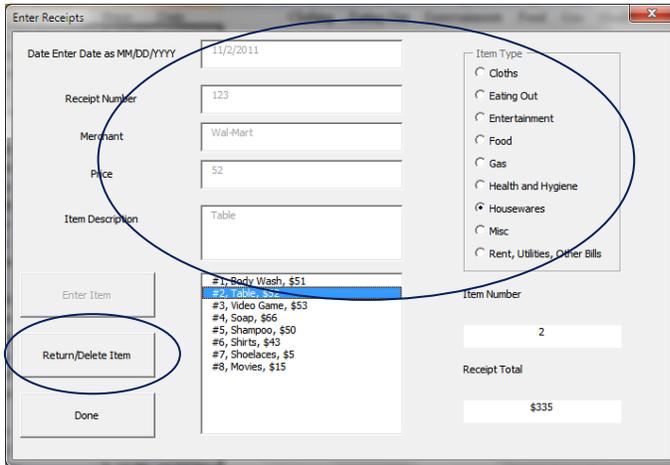
- c. Double clicking on the receipt number will reopen the Receipt form but some of the functionality has been disabled to prevent entering an item twice.
  - i. Accessing the receipt form through the list box disables all the text fields, the item category and the enter receipt button. The only buttons that are enabled are the return button and the "Done" button which returns the user to the welcome screen.
  - ii. When the user selects an item from the list, all fields are populated with the item information. This allows the user to review all the information about the item before deleting it. This also allows the user to review all items for a specific receipt even if no return is needed.



- d. To return an item, select the item from the list box and press the Return/Delete Item button. The program will then search through the worksheet to find the item

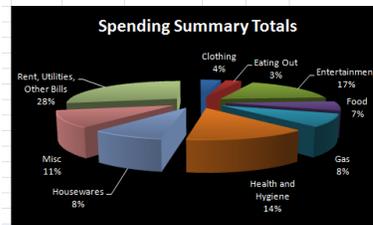
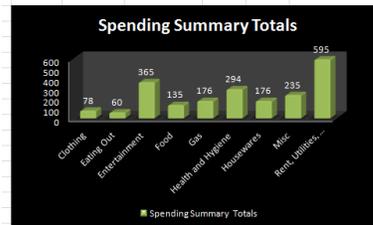
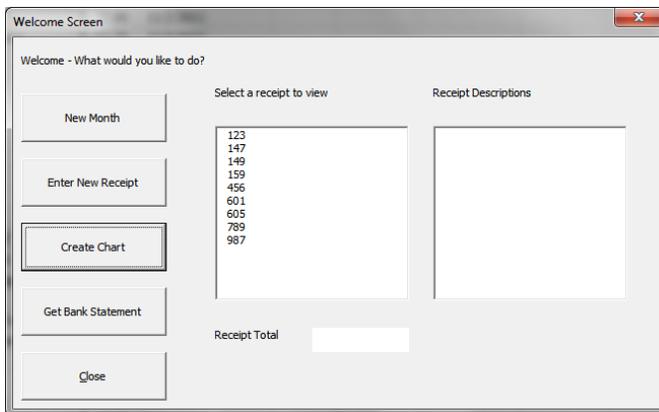
(using the item number described above), delete the item from the worksheet, and update all items on the worksheet and in the program to reflect the change.

- e. When the user presses the Return/Delete Item a message box asks the user to confirm the return. This prevents the user from accidentally deleting an item.



The table was returned and removed from the list and worksheet

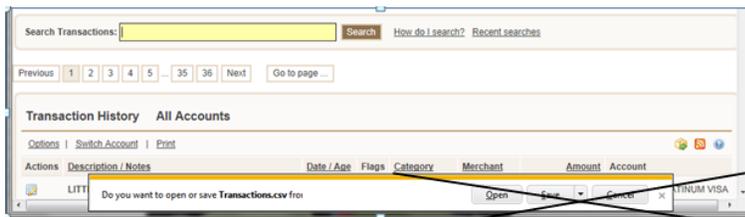
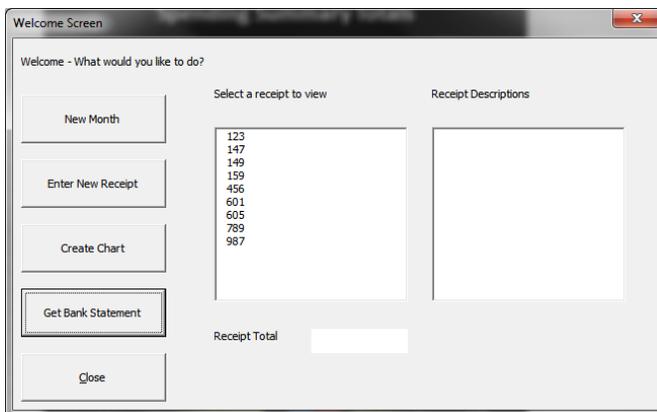
8. Creating Charts – The most useful evaluation tool in the program
  - a. One of the coolest parts of the program is its ability to create summary charts of the purchases by item category. This is a necessary part of the program because it allows the user to track and evaluate their spending habits.
  - b. To create a chart simply press the create chart button on the welcome screen. The program will then summarize all the data by item category and produce two charts (one for dollars spent, the other in percentages of total spending) and positions the charts in the middle of the sheet for easy viewing.



| Category                     | Amount      | Percentage |
|------------------------------|-------------|------------|
| Clothing                     | 78.00       | 4%         |
| Eating Out                   | 60.00       | 3%         |
| Entertainment                | 365.00      | 17%        |
| Food                         | 135.00      | 7%         |
| Gas                          | 176.00      | 8%         |
| Health and Hygiene           | 294.00      | 14%        |
| Housewares                   | 176.00      | 8%         |
| Misc.                        | 235.00      | 11%        |
| Rent, Utilities, Other Bills | 595.00      | 28%        |
| <b>Total</b>                 | <b>2114</b> |            |

9. Retrieving Bank Transactions. (Note – this feature was made specific for my computer. To get the functionality to work the user will need to change the URL's in the code to match their bank and they change the login tags. Additionally, the user will need to change the location of the downloaded file and the location the file is to be moved to.)
  - a. The final feature of the program is the Get Bank Statement feature
  - b. This feature goes onto the user's bank website and downloads a csv file containing all transactions for the year into the Downloads folder on the C drive.

- i. Before accessing the bank website a login form pops up asking the user to provide the bank login credentials.
  - ii. When IE opens DO NOT touch anything. The program is set to send a key stroke to IE to retrieve the file. If you navigate away from the page the program will crash. (Since I had to send a key stroke I had to allow the IE window to open at least for a little bit of time.)
  - iii. The program will log-out of the bank website to allow for future access
- c. The program moves this file into the workbook location (deleting any previous copy of this file)
- d. The program then imports the transactions into the worksheet and delete all transactions from the imported list that do not fall within 2 days of the oldest and newest transactions listed in column "G" of the worksheet. (This allows for any delays in transaction posting to the bank website.)
- i. Note – I have provided screen shots of the process but have edited the data to eliminate any confidential information



| Bank Statement |            |            |               |             |      |       |
|----------------|------------|------------|---------------|-------------|------|-------|
| date           | amount     | balance    | category      | description | Memo | Notes |
| 12/2/2011      | (\$550.00) | \$112.31   | DebitNone     |             |      |       |
| 12/2/2011      | (\$150.64) | (\$695.95) | Insurance     |             |      |       |
| 12/2/2011      | (\$48.00)  | (\$545.31) | Groceries     |             |      |       |
| 12/2/2011      | (\$34.69)  | (\$497.31) | DebitNone     |             |      |       |
| 12/2/2011      | (\$20.00)  | (\$462.62) | DebitNone     |             |      |       |
| 12/2/2011      | (\$8.54)   | (\$462.62) | Entertainment |             |      |       |
| 12/2/2011      | (\$3.52)   | (\$434.08) | DiningOut     |             |      |       |
| 12/1/2011      | (\$5.03)   | (\$430.56) | Groceries     |             |      |       |
| 12/1/2011      | (\$2.00)   | (\$425.53) | Utilities     |             |      |       |
| 11/30/2011     | \$0.23     | \$951.38   | Dividends     |             |      |       |
| 11/30/2011     | \$0.24     | \$1,707.65 | Dividends     |             |      |       |
| 11/30/2011     | (\$38.33)  | (\$423.53) | DebitNone     |             |      |       |
| 11/30/2011     | (\$17.98)  | (\$385.20) | Household     |             |      |       |
| 11/30/2011     | (\$8.00)   | (\$367.22) | Household     |             |      |       |
| 11/30/2011     | (\$29.25)  | \$662.31   | DebitNone     |             |      |       |
| 11/28/2011     | \$359.00   | (\$359.22) | Loan          |             |      |       |
| 11/28/2011     | \$75.00    | \$691.56   | CreditNone    |             |      |       |
| 11/28/2011     | \$75.00    | \$1,707.41 | CreditNone    |             |      |       |
| 11/28/2011     | (\$251.37) | (\$718.22) | Household     |             |      |       |
| 11/28/2011     | (\$170.75) | (\$466.85) | Technology    |             |      |       |
| 11/28/2011     | (\$89.69)  | (\$296.10) | Household     |             |      |       |
| 11/28/2011     | (\$60.79)  | (\$206.41) | Household     |             |      |       |
| 11/28/2011     | (\$53.43)  | (\$145.62) | Household     |             |      |       |
| 11/28/2011     | (\$18.26)  | (\$92.19)  | Groceries     |             |      |       |
| 11/28/2011     | (\$17.90)  | (\$73.93)  | Household     |             |      |       |
| 11/28/2011     | (\$5.22)   | (\$56.03)  | Medical       |             |      |       |
| 11/28/2011     | \$101.41   | (\$50.81)  | Interest      |             |      |       |
| 11/27/2011     | \$600.00   | (\$152.22) | Loan          |             |      |       |

This is an edited example of the query that is pulled into the worksheet

This concludes the user implementation Documentation.

### Learning and Conceptual Difficulties Discussion

This project provided a great opportunity to learn several new things about VBA and programming in general.

The biggest thing I learned during this project is how difficult it is to actually design a program to solve a problem. I really struggled identifying exactly what I wanted my program to do. Often what would occur is I would identify a few things I wanted the program to do and as I began programming I would realize I actually needed the program to include an extra feature or extra text field or I would need the program to act differently. This would cause me to have to stop what I was working on and change my code. This was not a big problem at first when the program was small but as I worked longer on the project the code got bigger and more difficult to change. I would have to spend a lot of time trying to change my code to include this forgotten feature when I could have saved a lot of time. Inevitably each change would result in many errors in my old code so I not only had to add the new feature but I had to fix the old code to work with it as well. If I had spent more time in the planning and design phases I would not have had to continually change my code and would have saved a lot of grief and time. I now know that I need to spend more time in the design and planning phases because it is easier to implement a feature at the beginning than it is at the end and this will save a lot of time making future programs.

Another thing I learned doing this project is how to interact with a secure website through VBA. One of the most difficult parts of the project was getting my program to log onto my bank website and download the CSV file with all my transactions. At first I could not get my code to pass my user name and password to the bank site so I had to go into the website source code and locate the correct tags. I then had to figure out how to navigate to the page I wanted so I could

access the downloadable file. I learned a lot about the internet agent class Dr. Allen provided in doing this. Some of the functions I discovered were the savefile function and the followlinkbytext or by href all of which I tried to use in my program. One interesting aspect I encountered is that if the bank updates its website to show special deals or be more festive then the tags will all change. To get around this problem I needed to find a generic login location on the website that was unlikely to change tags. Thus instead of going to the ever changing home page I went to a separate page (usually only used when you incorrectly enter login info) and used those tags and was able to successfully log into the bank website.

One major problem I encountered in the project was getting the csv file to download from the bank website. I was able to navigate to the correct URL but the savefile function did not download the transactions. With Dr. Allen's help I tried changing my IE security settings to download files automatically which was unsuccessful and ultimately had to learn how to send key strokes through VBA in addition to using the Application.wait function (another function I learned about) to add pauses in the program in order to successfully download the file. By sending alt + s (effectively telling the computer to download and save the file) through VBA and adding several pauses I was able to download the file to my hard drive. This presented a new challenge in that I needed to move the file from my downloads folder to the workbook location and delete any old instances of the csv file. To do this I learned to use the Kill function in VBA, which will delete a file if you give it the complete file path. I had to delete all the old files so the query would pull the most recent transactions. I also learned to use the FileCopy function, which moves a file from a given location to a new location you specify using file paths, in order to move the downloaded file from the downloads folder to my workbook location. Once I had moved the file it was simple to import the data through the query wizard and build this functionality into my program.

Another problem I encountered involved my charts. Using the macro recorder I was able to learn how to create charts and even how to change the format/style of the charts through changing the code. One issue I had is that I wanted to make multiple charts but in excel the charts are all placed in the same spot initially making it impossible to view more than the 1<sup>st</sup> chart. Now I could have easily just moved the charts by hand each time but since I was making a program I wanted this to happen automatically. To solve the problem I had to learn about chart objects. When I would record the macro, excel would give the charts a new name like chart 1 or 2. This name changed each time a new chart was made making it impossible to automatically copy and paste the new charts. Therefore, instead of having the program move the chart based on the chart name I had the program move the chart based on the chart object number. Each worksheet keeps a list of all the charts that exist in the worksheet using chart objects. I figured out that if I deleted the old charts before adding the new charts I could substitute the chart name for the chart object reference. This way there would only ever be two chart objects. Once I made this adjustment it was easy to tell excel which chart to move where because I always created the charts in the same

Ron Barwick  
MBA 614

order so I could always move them to the same place regardless of how many times I made new charts.

Another thing I learned in this project is the proper way to use the Instr function. In the past I had only ever used this function to tell if a character or string was in another string. I had never used it to actually get the location of the desired character. This is actually what the function is for. Using the Instr function was critical to parsing some of the strings in my list boxes to get the numbers from the string. I used commas to delineate the strings and used the Instr function to find the location of the first comma in the string. When I had this location, I then was able to use the Left function to collect the information before the comma and use that information to look up specific spreadsheet addresses. This was critical to using my Return/Delete feature, as I needed a way to locate a specific item from a specific receipt on the spreadsheet so I could delete it.

One last interesting thing I learned was the IsDate function. This function will take data and try to convert it to a valid date. This was useful as a field check for my date text box. This function is limited in that it will accept anything that looks like a valid date. For instance if I pass the function 13/12/2011 instead of December 13, 2011 ( or 12/13/2011) IsDate() will not flag this as an invalid date even though I intended for that the user enter the date in this format MM/DD/YYYY. Instead, it will read this as December 13, 2011 because DD/MM/YYYY is a valid date format as well. I handled this issue by using a form label to tell the user how to enter the date properly.

I really enjoyed working on this project and learned a lot about VBA and programming in general.