

IS520 Final Project

Ziyun Song

Brigham Young University

April 14, 2015

## **Project Proposal**

I will soon start working, and will be living on my own salary. I want to build a project that will manage and budget my personal finance. I will set a goal of how much I want to save each month. The project will automatically calculate my Federal, State, Local, Social Security, Medicare tax with holdings, and Net Pay based on my Gross Income Input.

Then after importing my credit card transaction file (.csv), the project will categorize each transaction and plot a graph on what percentage each category represents. It can also compare how much I budgeted and see if I went over or down. Then the project calculates how much I can put into my savings account, plot a graph that shows the trend of how much I saved each month, and compare my goal at the beginning of the month see if I met it.

The project can be executed by formulas and VBA codes that will automate the process of importing and analyzing csv files of credit card transactions.

## **Executive Summary**

This project runs personal finance analysis based on the inputs of current month's credit card transactions, before-tax (Gross) paycheck amount, expense budget, savings goal, and savings from last month.

After these data are collected from the user by a user form, the project runs calculations and analysis to output Federal, State, Local, Social Security, Medicare Withholdings, and Net Pay. The project also generates analysis of all incoming and outgoing card transactions, gives a summary of expense amount in each category. Then the project runs some Actual VS. Budget comparisons of your expenses, savings, and if your goals are met.

After the number analysis are generated, the user can choose to plot graphs based on the numbers, and this gives the user a more visual understanding of his/her personal finance condition of the current month. The user can also choose to clear contents and have the project ready for next month's use.

## Implementation

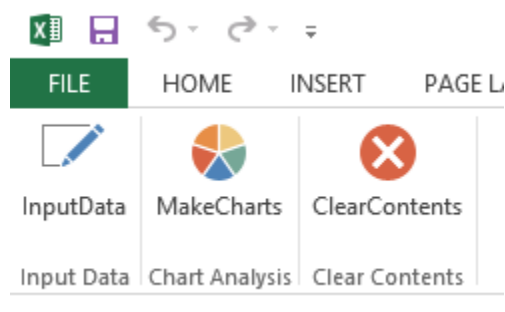
This Project contains the following components:

- 1) Data Input
- 2) Import Bank Statement from a different location
- 3) Calculate all kinds of withholdings based on IRS Data
- 4) Analyze and Categorize all credit card transactions
- 5) Print results to worksheet
- 6) Plot analysis charts to worksheet
- 7) Clear Contents

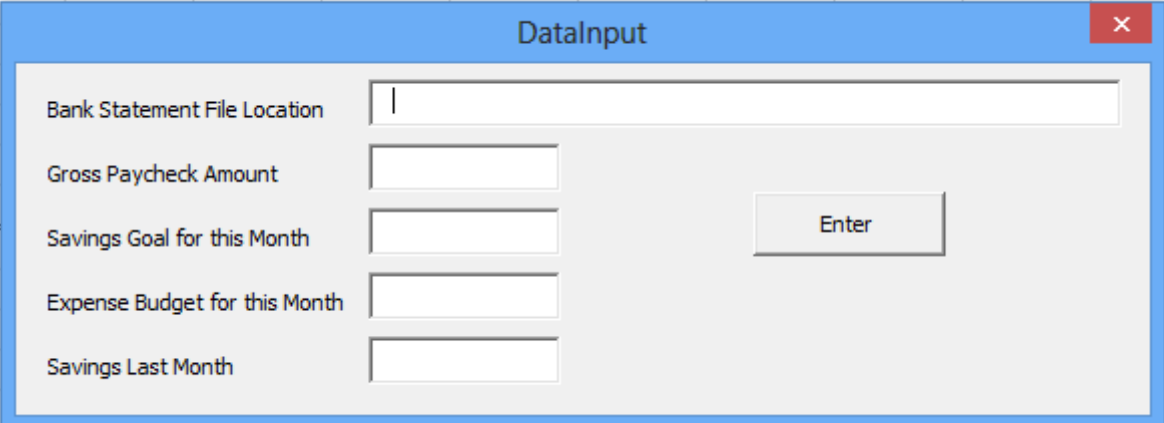
### 1) Data Input

This component is implemented by using data form and has a macro that attached to a customized ribbon to call the data form. This component is included because certain data inputs from the user are necessary to execute the project and using data form is a more user-friendly and convenient way.

The user will start the project by clicking the “Input Data” button under the “Run Project” Tab



Then a User Form Input Box will be populated to allow the user to input key data

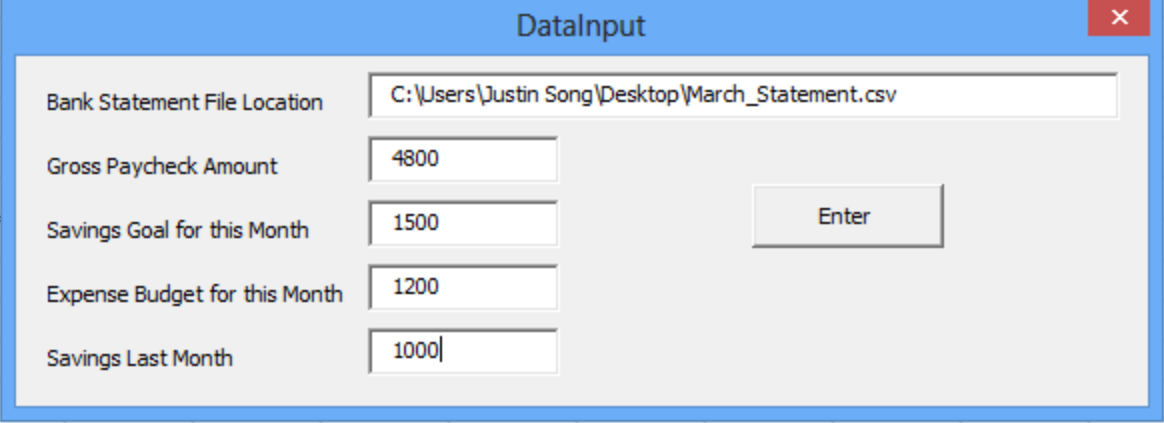


The screenshot shows a user form titled "DataInput" with a blue header and a red close button. The form contains five input fields arranged vertically on the left, each with a label to its left. To the right of these fields is a single "Enter" button. The input fields are currently empty, with a cursor visible in the first field.

Label	Input Field
Bank Statement File Location	
Gross Paycheck Amount	
Savings Goal for this Month	
Expense Budget for this Month	
Savings Last Month	

Enter

Sample Inputs as following:



The screenshot shows the same "DataInput" form, but now with sample data entered into all five input fields. The "Enter" button remains to the right of the fields.

Label	Input Field
Bank Statement File Location	C:\Users\Justin Song\Desktop\March_Statement.csv
Gross Paycheck Amount	4800
Savings Goal for this Month	1500
Expense Budget for this Month	1200
Savings Last Month	1000

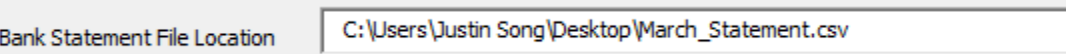
Enter

After the user hit "Enter" on the user form input box, the project will start.

## 2) Import Bank Statement From a Different Location

This component is implemented by using VBA codes to connect an outside file and copy the file into a newly added sheet named "Data". This component is included because bank transactions are needed to analyze the user's expenses.

The bank statement file location was one of the data inputs (See Data Input Section):



This is a close-up view of the "Bank Statement File Location" input field from the form. The text "C:\Users\Justin Song\Desktop\March\_Statement.csv" is clearly visible within the input box.

Label	Input Field
Bank Statement File Location	C:\Users\Justin Song\Desktop\March_Statement.csv

The import process is executed by VBA codes as following:

---

```
Dim CSV As Worksheet
Set CSV = ThisWorkbook.Sheets.Add(After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count))
CSV.Name = "Data"
With ThisWorkbook.Sheets("Data").QueryTables.Add(Connection:=connectionName, Destination:=Range("$A$1"))
    .Name = "Statement"
    .FieldNames = True
    .RowNumbers = False
    .FillAdjacentFormulas = False
    .PreserveFormatting = True
    .RefreshOnFileOpen = False
    .RefreshStyle = xlOverwriteCells
    .SavePassword = False
    .SaveData = True
    .AdjustColumnWidth = True
    .RefreshPeriod = 0
    .TextFilePromptOnRefresh = False
    .TextFilePlatform = 437
    .TextFileStartRow = 1
    .TextFileParseType = xlDelimited
    .TextFileTextQualifier = xlTextQualifierDoubleQuote
    .TextFileConsecutiveDelimiter = False
    .TextFileTabDelimiter = False
    .TextFileSemicolonDelimiter = False
    .TextFileCommaDelimiter = True
    .TextFileSpaceDelimiter = False
    .Refresh BackgroundQuery:=False
End With
```

### 3) Calculate All Kinds of Withholdings Based on IRS Data

This component is implemented by using VBA codes to calculate withholdings based on IRS Data. Because every category of withholdings has different income brackets, thus, multiple nested IF statements are used to complete this component. This component is included because it is useful to show the user how much was deducted from his/her paycheck for taxes, and gives the user a Net Pay amount that is available to use for the month.

Sample VBA codes to calculate State Withholding:

```
If GrossPay >= 42917 Then
    StateWithheld = (GrossPay - 42917) * 0.0962 + 3990.46
ElseIf GrossPay >= 40871 And GrossPay < 42917 Then
    StateWithheld = (GrossPay - 40871) * 0.4902 + 2987.42
ElseIf GrossPay >= 10215 And GrossPay < 40871 Then
    StateWithheld = (GrossPay - 10215) * 0.0735 + 734.23
ElseIf GrossPay >= 8173 And GrossPay < 10215 Then
    StateWithheld = (GrossPay - 8173) * 0.0815 + 567.77
ElseIf GrossPay >= 6129 And GrossPay < 8173 Then
    StateWithheld = (GrossPay - 6129) * 0.0715 + 421.62
ElseIf GrossPay >= 4085 And GrossPay < 6129 Then
    StateWithheld = (GrossPay - 4085) * 0.0808 + 256.42
ElseIf GrossPay >= 3675 And GrossPay < 4085 Then
    StateWithheld = (GrossPay - 3675) * 0.0758 + 225.38
ElseIf GrossPay >= 3062 And GrossPay < 3675 Then
    StateWithheld = (GrossPay - 3062) * 0.0665 + 184.62
ElseIf GrossPay >= 813 And GrossPay < 3062 Then
    StateWithheld = (GrossPay - 813) * 0.0645 + 39.58
ElseIf GrossPay >= 529 And GrossPay < 813 Then
    StateWithheld = (GrossPay - 529) * 0.059 + 22.81
ElseIf GrossPay >= 446 And GrossPay < 529 Then
    StateWithheld = (GrossPay - 446) * 0.0525 + 18.46
ElseIf GrossPay >= 323 And GrossPay < 446 Then
    StateWithheld = (GrossPay - 323) * 0.045 + 12.92
Else: StateWithheld = 0
End If
```

#### **4) Analyze and Categorize all credit card transactions**

This component is implemented by using VBA codes to calculate the amount of each expense category. The code will determine the last row of the “Data” Sheet, and then a For Loop with nested IF Statements is used to complete the component. This component is included because it is useful to show the user what amount is expensed in each category this month.

VBA code to complete this component:

```
ActiveWorkbook.Sheets("Data").Range("A1").Select
With Range("A1")
    LastRow = Range(.Offset(0, 0), .End(xlDown).End(xlToRight)).Rows.Count
End With
Dim i As Integer
For i = 1 To LastRow Step 1
    ActiveWorkbook.Sheets("Data").Select
    If Cells(i, 3).Value = "Gas" Then
        Gas = Gas + Cells(i, 2).Value
    ElseIf Cells(i, 3).Value = "Restaurant" Then
        Restaurant = Restaurant + Cells(i, 2).Value
    ElseIf Cells(i, 3).Value = "eCheck" Then
        eCheck = eCheck + Cells(i, 2).Value
    ElseIf Cells(i, 3).Value = "Miscellaneous" Then
        Miscellaneous = Miscellaneous + Cells(i, 2).Value
    ElseIf Cells(i, 3).Value = "Transfer" Then
        Transfer = Transfer + Cells(i, 2).Value
    ElseIf Cells(i, 3).Value = "Hotel" Then
        Hotel = Hotel + Cells(i, 2).Value
    ElseIf Cells(i, 3).Value = "Retail" Then
        Retail = Retail + Cells(i, 2).Value
    ElseIf Cells(i, 3).Value = "Grocery" Then
        Grocery = Grocery + Cells(i, 2).Value
    ElseIf Cells(i, 3).Value = "Airline" Then
        Airline = Airline + Cells(i, 2).Value
    ElseIf Cells(i, 3).Value = "Rent" Then
        Rent = Rent + Cells(i, 2).Value
    ElseIf Cells(i, 3).Value = "Return" Then
        PReturn = PReturn + Cells(i, 2).Value
    ElseIf Cells(i, 3).Value = "Direct Deposit" Then
        DirectDeposit = DirectDeposit + Cells(i, 2).Value
    Else: Other = Other + Cells(i, 2).Value
    End If
Next
```

## 5) Print results to worksheet

This component is implemented by using VBA codes to print calculation results into the worksheet, so the user can see them. The results will show immediately after the user enters data inputs. This Component is included because it is necessary to let the user see all calculation and analysis results in a nicely formatted and user-friendly way.

Sample Print Results:

### Paycheck Analysis

	PayCheck Analysis
GrossPay	4,800.00
Federal Withheld	1,028.49
State Withheld	314.19
Local Withheld	180.49
Social Security Withheld	297.60
Medicare Withheld	69.60
Net Pay	2,909.63

### Transaction Analysis

	Transaction Analysis
Direct Deposit	2,886.74
Transfer	235.24
Purchase Return	602.16
<u>Total Incoming</u>	3,724.14
Gas	51.99
Restaurant	320.63
eCheck	20.00
Miscellaneous	13.94
Hotel	337.44
Retail	240.30
Grocery	96.12
Airline	484.20
Rent	397.00
Other	-
<u>Total Outgoing</u>	1,961.62



## Budget VS. Actual Analysis

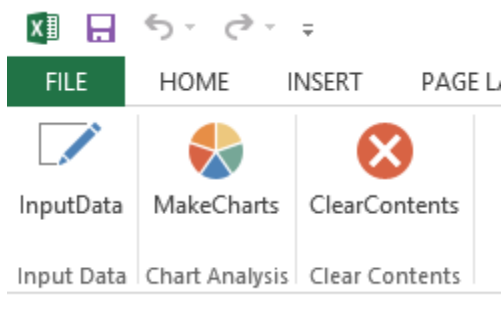
	<b>Expense Actual VS. Budget</b>
<b>Actual Expense (Minus Purchase Return)</b>	1,359.46
<b>Budget Expense</b>	1,200.00
	<b>Saving Goal VS. Current VS. Last</b>
<b>Goal</b>	1,500.00
<b>Saving This Month</b>	1,550.17
<b>Saving Last Month</b>	1,000.00
	<b>Results</b>
<b>Expense</b>	<b>Went Over Budget 159.46</b>
<b>Saving</b>	<b>Beat Goal By 50.1680000000001</b>

### 6) Plot charts of analysis to worksheet

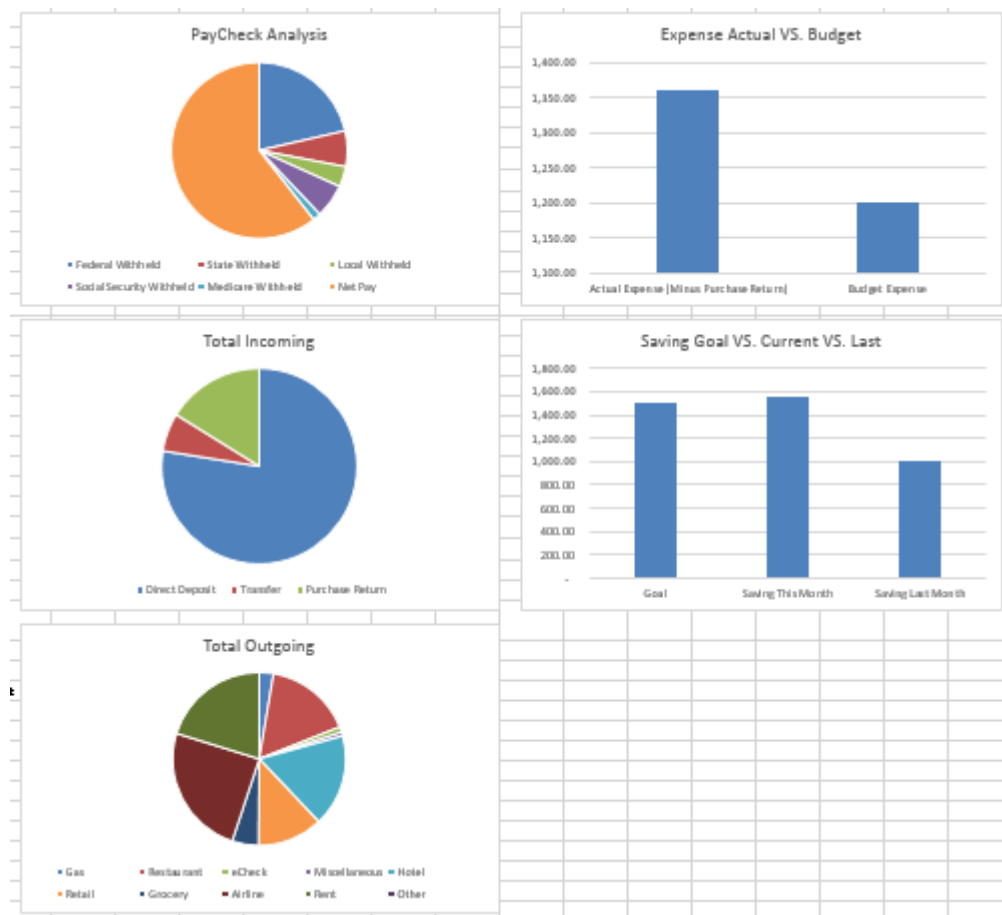
After the user sees the analysis result, the user have the option to click the “Make Charts” Button to make charts based on the numbers, thus generates a more straight-forward understanding of his/her Pay Check, expenses, and budgets.

This component is implemented by using VBA codes to draw pie charts or bar charts automatically based on labels and data. This Component is included because it gives a more straightforward and user-friendly view of the analysis.

Button:



## Sample Charts:



## Sample VBA Codes:

'To Make Pie Chart

```
Function PieChart(l As Range, d As Range, t As String, tp As Single) As Single
```

```
    Dim sh As Shape
    Set sh = ActiveSheet.Shapes.AddChart2(251, xlPie)
    sh.Chart.SetSourceData Source:=Union(l, d)
    sh.Chart.HasTitle = True
    sh.Chart.ChartTitle.Text = t
    sh.Top = tp
    PieChart = sh.Height
```

End Function

'To Make Bar Chart

```
Function BarChart(l As Range, d As Range, t As String, tp As Single, lft As Single) As Single
```

```
    Dim sh As Shape
    Set sh = ActiveSheet.Shapes.AddChart2(201, xlColumnClustered)
    sh.Chart.SetSourceData Source:=Union(l, d)
    sh.Chart.HasTitle = True
    sh.Chart.ChartTitle.Text = t
    sh.Top = tp
    sh.Left = lft
    BarChart = sh.Height
```

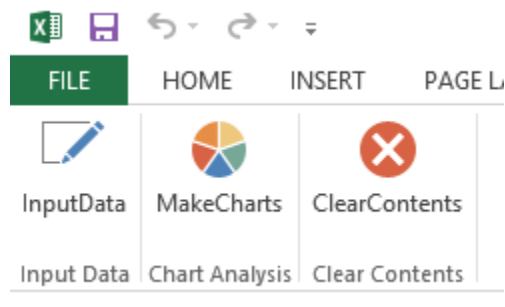
End Function

## 7) Clear Contents

After the user finishes reviewing all analysis and charts. The user have the option to click the “Clear Contents” button to clear all contents and have the project ready for next time’s use.

This component is implemented by using VBA codes to clear contents of certain ranges, delete the “Data” sheet, and delete all charts that are created. This component is included because it allows the user to have the project ready for next time’s use.

Button:



Sample VBA Codes:

```
Sheets("Data").Delete

Dim x As Integer
Dim s As Worksheet

Set s = Sheets("Analysis")

For x = s.Shapes.Count To 1 Step -1
    If s.Shapes(x).HasChart = True Then
        s.Shapes(x).Delete
    End If
Next
```

## **Discussion of Learning**

Concepts learned/reviewed in this project:

- Recording/Modifying Macro
- Customize Ribbons\*
- User Forms\*
- Charts\*
- Variables/Methods/Objects/Properties/Ranges/Sheets/Workbooks/etc.
- Subs/Functions
- Nested IF Statements
- For Loops
- Debug

There weren't any particular concepts too difficult to implement, concepts marked \* are the ones relatively difficult.

## **Assistance**

I did not receive substantial assistance throughout the project.