

Aluminum Rings Database

A VBA project for FLIE

By Michael Hsiung

Executive Summary

Description of the Company

FLIE, LLC is a Provo-based start-up that focuses on millennial based fashion. While the eventual goal of the company was to expand into other fashion accessories that could be sold at big retailers such as PacSun and Tilly's, the founder wanted to find a fashion accessory that the target market could quickly embrace and cause future brand awareness. The owner decided that fashion customized aluminum rings the right niche to launch FLIE into young fashion retail.

A Kickstarter was launched, more than 400 sales were made, and more than \$10,000 was collected. After survey information for all the backers of the Kickstarter were collected, Ty needed to start production for the new rings. Several problems arose as the production process started. First problem: ring sizing is very difficult without a first-hand experience, so the customers are constantly changing the order before the production process could be solidified. Second problem: with seven different colors, sixteen different sizes, and two different ring shapes, FLIE has to deal with 224 unique inventory variety. To complicate the process, the batch process of manufacturing requires two separate manufacturing process: molding and anodizing. The molding requires the production plain uncolored aluminum rings of 16 different sizes, and the anodizing process requires 7 separate batches of different colors.

Solution

The solution used in this Visual Basics Application (VBA) project is to create an easy to use editor for the data base, and create a statistics spreadsheet that display the different preferences for color, sizes, and shape. The use of an editor is produced in order to address the first problem of the project which is the frequent changing of the user's order. The editor allows the data base to stay consistent and uniform with the method and style of entering. Important data points such as color, size and shape is entered using a combo box instead of manually, in order to prevent discrepancies and misspelling of words.

The second problem is addressed through the statistical aggregator that I have design using VBA. Usually, a problem like this could be solved using a pivot table. Due to poor foresight, the survey author did not anticipate the future problems that he would face the way it is designed. First, the survey combined shape type and ring size into one column only separated a hyphen i.e. "THIN – 7.5". In addition to the need for parsing data, the data is sorted by persons. Therefore, each purchase has a different number of rings of different sizes further making the data difficult to sort when rings are in four different columns.

For more information regarding the Kickstarter and Video:

<https://www.kickstarter.com/projects/flie/flie-rings-colorful-anodized-aluminum-rings-for-li>

For more information regarding FLIE LLC.:

<https://flielife.com/>

Implementation Documentation

Data Editor

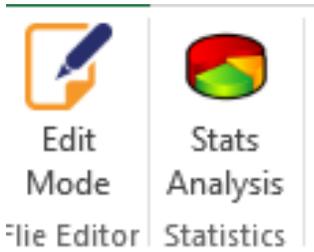


Figure 1: The two buttons used to access the sub

The project is broken into two parts (see figure 1). The first tool designed is called “Edit Mode” which allows the user to edit individual lines of data. The editing tool is a user form generated via VBA code. The first thing required of the project is to create the form using the design mode provided in the VBA Excel program. The user form is organized so customer information is aligned on the top with text boxes displaying the user name. Except for engraving, which is a combo box with two choices. On the lower half of the user form, the order information is ordered from the first order to the fifth order with the three specifications listed: color, type, and size. At first, I wanted to make the text boxes really small so the user form would not have to be as big, but the texts did not fit inside my desired size. So, I had to increase the size of the overall user form in order to not decrease the size of the text.

In the user form, many of the combo boxes require preloaded lists of items of choice (see figure 2). In order to load them, I would need to create two sets of loops for each type of upload. The first type of loop is to upload all the possible selection of choices into an array. For example, I needed an array of variables that stores all the possible colors, so I would load into the string array “color” with all the string names associated with the 7 different color choices. The second type of loop is to add all of the lists created from the arrays from the first type of loop and attach them to the appropriate combo boxes. At first, I manually attached each detail of each of the arrays into the individual combo boxes. I then realized that the process would be much faster if I could create an array with the object type as combo boxes so they could attach all the items in a loop (see frmEdit/sub objectupload).

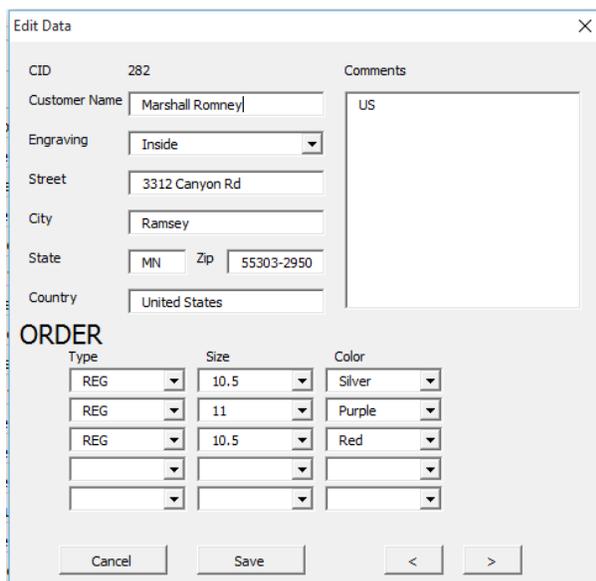
The screenshot shows a window titled 'Edit Data' with a close button in the top right. The form contains several text boxes and dropdown menus. At the top, 'CID' is 282 and 'Comments' is 'US'. Below that, 'Customer Name' is 'Marshall Romney'. 'Engraving' is a dropdown menu set to 'Inside'. 'Street' is '3312 Canyon Rd', 'City' is 'Ramsey', 'State' is 'MN', and 'Zip' is '55303-2950'. 'Country' is 'United States'. Below this is a section titled 'ORDER' with three columns: 'Type', 'Size', and 'Color'. Each column has a dropdown menu. The first row shows 'REG', '10.5', and 'Silver'. The second row shows 'REG', '11', and 'Purple'. The third row shows 'REG', '10.5', and 'Red'. There are two empty rows below. At the bottom are 'Cancel', 'Save', and navigation arrows.

Figure 2: User form layout

After the user form is designed, I began programming the initialized response of the user form. When the user form is called, I wanted to excel to load all the content of the active cell's row into the user form's appropriate locations. A loop is designed so upon loading the “Edit Mode” sub procedure, the first thing the code does is to grab everything on the line and assign each column into part of an array. The code then assigns the appropriate value into the appropriate textbox.

Please note that columns D, F, H, J, and L each splits into two separate entries. The split function is used to attach the two types of variables into separate combo boxes. Since there is a consistent “space-hyphen-space” between ring shape and ring size, I have use the split function using “ – “ as the delimiter. While designing the loop, I would into the problems when try running the sub procedure if not all five ring slots are filled. This problem is caused by the fact if the split function is used to assign variables into an array. If one of the split column is empty on excel, no data would be assigned to the array. When the code starts referencing splitter(0) or splitter(1), the program stops. Therefore, the “if” function is used to prevent that issue and would enter empty data into the combo boxes.

The last part of the user form is the command buttons. Four buttons are created: cancel, save, backwards, and forward. The code of cancel simply closes the user form. The save button when click will trigger changes in the user form to be replaced into the excel spreadsheet. The difficult function of the save button is preventing mistakes in edit. Because the purpose of this project is to maintain data integrity, the lower half of the form goes through a check point before the data in imprinted on excel. The checkpoint checks to only allow the code to pass through if (1) all of the data in one row if not empty or if (2) all of the data in one row is empty. If that is not the case, the “if” statement would return a message box letting the user know of their errors and ask them to fix it. The Boolean logic took me a really long time to figure out, but now it works. The forward and backwards buttons are similar to the save button except for the ability to save. The same type of checkpoint is used to prevent the user from moving on to the next line.

Stats Analysis

This macro tallies all the data points of interest and project the new data in a new spreadsheet. Two types of data is being collected in this project. One is the individual counts of every possible attribute, and the other is the individual count of every possible combination rings. Finally, the sub procedure seeks to present the data, especially the second type, in a easy to read sheet.

The sub procedure is set up to do three things in order:

- (1) Upload the relevant comparisons of attributes into an array
- (2) Create a do loop that compares one line at a time and offsets
- (3) Print the results and findings into a spreadsheet.

Ring Colors	Count
Black	103
Silver	45
Gold	24
Red	75
Blue	77
Green	45
Purple	49

Uploading the relevant comparisons did not result in much trouble. The loop is extremely similar to the same loop performed in the “Edit” sub procedure. The second do loop a little more difficult. Instead of creating an array, I decided to use the mid and left function to compare the ring sizes with the preassigned choices. If there is a match the loop will tally the attribute of interest.

Figure 3: An example of a tally

What makes the loop more difficult is combining the three attributes together and making a variable for all 224 combination of possible rings. I had to think about this for a very long time, and I have decided to use a three dimensional array to allocate a spot for each possible combination. I made all of the individual combination unique by concatenating all the possible combination of the strings in order. Then I run a three nested loop to tally up all the possible combinations.

The last part of the project is to display the tally on a new sheet. The first type of data is easy to display in two columns. The second type of data display required a little more thinking. Since the data exists in three dimensions, I have tweak the two dimensional display to display all three attributes in matrix form. I have chosen to list all the colors in twos with on regular and one thin on the columns and listing all of the sizes on each row. A loop is created so all the appropriate count are printed on the appropriate matrix location. A conditional format is placed over the matrix to better visualize the trend and differences among product characterizations.

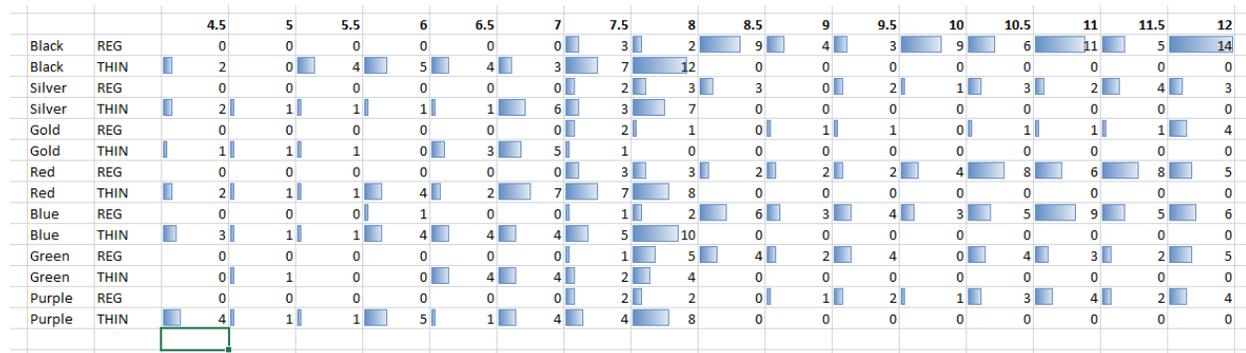


Figure 4: The REG rings are more often the larger sizes

Analysis of Interesting Trend

A couple interesting trend is found in your rings. There seems to be strong bimodal distribution in terms of ring size from this Kickstarter. My hunch seems to want to explain this by the difference between the male and female hand size. Unfortunately, gender identifiers from the survey, I only have names to use to confirm my suspicions. According to the conditional formatting, people with larger hands tend to favor regular shaped rings and people with smaller hands prefer thin shaped rings (see figure 4). If my hypothesis are correct, the preference may be caused by the differences in gender.

Assistance

I have received no major assistance from anyone. I have discussed some of my issues with my computer science roommate, Jeremy in regards to issues especially with uncleared memory. He was not much of a help since he was not familiar with VBA.

Privacy Statement

All of the names and street addresses have been replaced with different names and street addresses. Please only use the data set located in the excel sheet as educational purpose.