

Executive Summary:



Utah Valley Fencing Academy – Utah Valley Fencing Academy, Inc.(UVFA or the Academy) is a recently-registered non-profit organization in Utah Valley dedicated to teaching the art and sport of fencing and to prepare students of all ages to compete in fencing competitions. The organization itself has been around only since July of 2010 and is currently seeking 501(c)(3) non-profit status with the IRS. One of the most difficult parts of the application for the organization has been its inventory. The academy has an inventory of over \$2,000 with items ranging from replacement sword parts of less than \$2 a piece to fencing attire which can be over \$150 a piece.

Normally the inventory record keeping would not be so hard, except that inventory is constantly changing within the organization. Parts are constantly being replaced, equipment is being lent out to students for competitions, new inventory is being bought, and some inventory is being sold. Since inventory accounts for the majority of the academy's assets, they are required to keep a strict record of their inventory and provide accurate inventory numbers to meet the requirements of the 501(c)(3) application.

The UVFA Inventory System was created to provide the Academy with a simple tool to maintain the inventory records of the organization. The inventory systems shown here has the ability to track, search, and adjust inventory; account for depreciation; track inventory items issued to students; and track items purchased by students. The Excel XML ribbon was also modified to include a new UVFA Inventory table with custom buttons to allow easy access to the features and calculators of the inventory system. The financial figures and calculators offered in the system have must custom designed to match the accounts and records being maintained by the Academy in their QuickBooks accounting records.

* As you view the data in this report and the excel file, it should be noted that the system has not been fully implemented at UVFA yet and so some areas were blank. To facilitate testing and provide examples, test data was included that is completely made-up and does not reflect real people or figures.

UVFA Inventory Table & Menu:

When the macro-enabled workbook is opened, it is designed to start on the “Inventory” table. This table contains all of the inventory items that currently belong to UVFA and all information pertaining to these items in a raw format.

Date	Vendor	Arms	Category	Type	Condition	Size	Gender	Hand	Cost	Depreciation	Count	Description
8/4/2010	Absolute Fencing Gear, Inc.	Weaponry	Blanel	Foil	New				\$ 22.95	\$2.87	9	87904 - French F
8/4/2010	Absolute Fencing Gear, Inc.	Weaponry	Blanel	Epee	New				\$ 15.30	\$1.91	6	97904 - French F
8/4/2010	Absolute Fencing Gear, Inc.	Weaponry	Wire	Epee	New				\$ 3.19	\$0.40	1	97908 - French E
8/4/2010	Absolute Fencing Gear, Inc.	Misc	DVD	N/A	New				\$ 35.00	\$4.38	1	133029 - Referee
8/4/2010	Absolute Fencing Gear, Inc.	Attire	Jacket	N/A	New	42	Male	Left	\$ 43.20	\$5.40	1	21025-L42 AF Me
8/24/2010	Absolute Fencing Gear, Inc.	Attire	Mask	Foil	New	Medium			\$ 36.00	\$4.50	1	11001-M AF Stam
8/24/2010	Absolute Fencing Gear, Inc.	Attire	Plants	N/A	New	32			\$ 51.20	\$6.40	2	21023-32 AF Un
8/24/2010	Absolute Fencing Gear, Inc.	Attire	Jacket	N/A	New	32	Male	Right	\$ 36.00	\$4.50	1	21005-R32 AF Co
8/31/2010	Big 5 Sporting Goods	Misc	Pull-up Bar	N/A	New				\$ 32.04	\$4.01	1	Exercise pull-up
9/13/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Cord	N/A	New				\$ 83.20	\$10.40	8	BG 2-Prong class
9/13/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Socket	Foil	New				\$ 38.40	\$4.80	8	BG 2-Prong SOCK
9/13/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Base	Epee	New				\$ 25.50	\$3.19	10	French epee BAS
9/13/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Spring	Epee	New				\$ 5.61	\$0.70	20	French epee Cont
9/13/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Spring	Epee	New				\$ 5.61	\$0.70	20	French epee Pres
9/13/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Tip	Epee	New				\$ 26.40	\$2.55	6	French epee TIP
9/13/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Screw	Epee	New				\$ 5.60	\$0.70	20	SG epee SCREW
9/13/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Base	Foil	New				\$ 21.25	\$2.66	10	ST French foil BA
9/13/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Spring	Foil	New				\$ 2.81	\$0.35	10	ST French foil SP
9/13/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Tip	Foil	New				\$ 17.85	\$2.23	6	ST French foil TIP
9/13/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Wire	Foil	New				\$ 19.13	\$2.39	9	French Foil WIRE
9/15/2010	Burlington Coat Factory	Misc	Mats	N/A	New				\$ 102.45	\$12.81	12	Yoga Mats
9/24/2010	Utah Swords Academy	Weaponry	Blade	Sabre	New				\$ 225.00	\$28.13	5	Sabre blades
10/2/2010	Utah Swords Academy	Weaponry	Sword	Sabre	New	#5			\$ 315.00	\$55.13	7	Sabre # 5
10/2/2010	Utah Swords Academy	Attire	Mask	Sabre	New	Small			\$ 114.00	\$19.95	2	Saber Masks
10/2/2010	Utah Swords Academy	Attire	Cuff	N/A	New				\$ 36.00	\$6.30	3	Glove Cuff
10/5/2010	Blue Quantlet Fencing Gear, Inc.	Attire	Jacket	N/A	New	36	Male	Right	\$ 36.80	\$6.44	1	BG 350N Standar
10/5/2010	Blue Quantlet Fencing Gear, Inc.	Attire	Lame	Sabre	New	38			\$ 152.15	\$26.63	1	EDGE Stainless
10/5/2010	Ross	Misc	Mat	N/A	New				\$ 42.89	\$7.47	5	Yoga Mats
10/11/2010	Blue Quantlet Fencing Gear, Inc.	Weaponry	Socket	Epee	New				\$ 28.80	\$5.04	6	BG Epee SOCKE

Figure 1 – Inventory Table as visible after opening

Users are welcome to view and edit data in this format, but the inventory system includes several tools that can help a user more quickly locate, edit, or add items to this list. These tools are located on the Excel ribbon under a custom tab labeled “UVFA Inventory.”

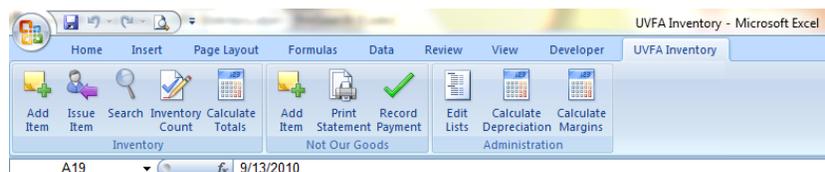


Figure 2 – UVFA Inventory Tab with custom buttons

The buttons on the UVFA Inventory tab are divided into three clusters. The first cluster is titled “Inventory” and contains all of the tools designed to maintain inventory items that belong to UVFA. The second cluster is titled “Not Our Goods” and contains tools that help the Academy maintain items that are purchased by the Academy for resale to students. Finally, the third cluster contains administrative tools that help the user customize options in the inventory system and also process annual depreciation calculations.

Inventory:

Add Item - The first tool available to the user is the option to add a new item to inventory. This is a simple user form similar to the one used in the in-class inventory example. The form has been designed in a format that mimics the format and information normally found on the invoices of the Academy's most popular suppliers. The vendor, category, type, and condition pick boxes are all filled based on menu lists maintained in a hidden table called Menu. These lists are customizable using a button in the administration cluster that will be covered later. These inputs, together with the date, count (quantity) and cost are all required fields that include message boxes requesting any missing information and will exit the add item procedure if incomplete. All other inputs are option as they do not appear on every item, but do help us categorize and track some items (such as clothing) better. The description line typically contains the item number or line information from the invoice and helps in placing future orders.

The screenshot shows a window titled "Add Inventory" with a close button in the top right. The form contains the following fields and controls:

- Vendor: Blue Guantlet Fencing Gear, Inc. (dropdown)
- Date: 12/08/2010 (text input)
- Category: Lame (dropdown)
- Type: Foil (dropdown)
- Condition: New (dropdown)
- Count: 1 (text input)
- Total Cost: 62.50 (text input)
- Size: XXXL (text input)
- Gender: Male Female (radio buttons)
- Hand: Right-Hand Left-Hand (radio buttons)
- Description: BG High Density copper foil lame, size XXXL (text input)
- Footer: (Size, Gender, and Hand are optional items)
- Buttons: Save, Cancel

Figure 3 – Add Inventory User Form

After included all of the necessary information, pressing the "Save" button will move through the inventory to the last written line, and then move one row down to ensure it is at the bottom of the list. The form then writes the information to the corresponding row adding the item to inventory.

Issue Item – The "Issue Item" form is actually two forms in one. The purpose of this form is to track existing inventory that has been issued to students, and to issue new inventory to students for their use at tournaments or other events. The top right and bottom sections of the form are for issuing items to users. In the top right, a user can input the information about the student that is borrowing the items, the date the items are to be returned, and what items are being borrowed. As the user enters item information and presses the "Issue" button, the items are added into an array and displayed in the list-view below. The array is two dimensional and the list-view contains multiple columns using the "ColumnCount" property to allow for all of the information to be displayed and stored in a single location. If the user closes the form at this point without printing an invoice, all data will be lost as they are only stored in the array and the

list-view. Once all items are entered, the user presses the “Print Invoice” button. When this button is pressed, the data from the array is transferred to the first blank row (and subsequent rows as necessary) at the bottom of the Issue table. These values are then added into the Issue Invoice table which is a special table designed to look like an invoice and contains all of the necessary agreements for borrowing items. The system calculates the values of each item taking into account current recorded depreciation and displays the name of the borrower and the date the items are to be returned. This invoice can be printed, signed, and saved until the items are returned. Prior to this system, the Academy had no written records of outgoing inventory which made performing audits and accounting for items very difficult.

Figure 4 – Issue & Return Form

Utah Valley Fencing Academy, Inc.
 720 South State Street
 Orem, UT 84058
 (801) 226-2401
 http://www.fencingutah.org

INVOICE
 Student Issued Inventory

Student: Robin Hood

Invoice Date: 1/1/2010
 Return Date: 1/25/2010

Line #	Category	Type	Size	Gender	Hand	Value
1	Jacket	Epee	42	Male	Right	\$145.00
2	Glove	Foil	10	Male	Right	\$145.00
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
2						\$290.00

As the above named student (or legal guardian of the above named student if student is under the age of 18), I take full responsibility for all equipment listed in this invoice until such time that it is returned to Utah Valley Fencing Academy, Inc. (hereafter known as "The Academy") in its original condition to The Academy on or before the return date above unless an agreement is made in writing with the director or coach of The Academy to return the items at a later date. I further agree that all items will be returned in their current condition. Any damage to the items on issuance will be noted on this invoice. I agree to pay The Academy for any items lost or damaged while in my care.

Student/Guardian: _____

Utah Valley Fencing Academy, Inc. _____

Figure 5 – Issued Items Invoice

Once items are returned, the top left section of the form can be used to return items. To declare an item returned, a user can use the previous and next buttons to move through the Issue table to find the item, or if they have found the item in the Issue table already, they can simply key in the line number and press the “Go” button to jump there. Once on the line, pressing the “Return” button will permanently remove the item. It was originally planned that the items would only be tagged as returned, but it was later decided that since the organization would be keeping copies of the invoices, and the items would be returned and checked for damage prior to flagging them in the system, there was no need to keep the records after that point and instead it was decided to remove the records immediately. It should be noted here, that the items being removed are just the issuance records kept in the Issue table. The original inventory records for these items are still maintained in the Inventory table. In addition to returning items, the top left section can be used to reprint invoices that may have been lost. To reprint a customer’s invoice, use the above mentioned navigation buttons to navigate to one of the items in the group to be reprinted and then press the “Reprint Invoice” button. An invoice like the one described previously will be printed for all items that share the same student name and return date recorded.

Search – The “Search” button allows a user to search the inventory in two different ways. First, a user can search for individual items. When this option is selected, the search form will move down the list row by row until it comes to an item that matches the search criteria. Each search criteria is optional and the form will only search for those criteria with values in them. After stopping on a matching item, a User can view and edit the information about that item by pressing the edit button.

The screenshot shows a window titled "Search Inventory" with a close button in the top right corner. Inside the window, there are two radio buttons: "Search For Single Item" (which is selected) and "Create Results List". Below these are several input fields: "Vendor:" (a dropdown menu), "Date:" (a text box), "Category:" (a dropdown menu with "Spring" selected), "Condition:" (a dropdown menu with "New" selected), "Type:" (a dropdown menu with "Epee" selected), and "Description Contains:" (a text box with "Contact" entered). At the bottom of the window, there are three buttons: "Edit", "Search", and "Cancel".

Figure 6 – Search Inventory Form

Figure 7 – Edit Inventory Item Form

The second search option is to create a results list. This option will search through all matching items and copy them over to the InvReport table which is configured to be a printable custom inventory list generated from the User’s search criteria. The report will display the search criteria on top, and a list of all items that match those criteria below.

Category	Type	Size	Gender	Hand	Quantity	Avrg. Value	Total Value
Blade	Sabre				5	\$ 39.37	\$196.87
Sword	Sabre	#5			7	\$ 37.12	\$259.87
Mask	Sabre	Small			2	\$ 47.03	\$94.05
Lame	Sabre		38		1	\$ 125.52	\$125.52
Blade	Sabre				3	\$ 16.50	\$49.50
Glove	Sabre				8	\$ 8.25	\$66.00
Sword	Sabre	#5			7	\$ 37.12	\$259.87
Sword	Sabre	#2			5	\$ 37.12	\$185.62

Figure 8 – Custom Inventory Report

If neither of the search options is selected prior to pressing the search button, the user will see a message reminding him to select a search option.

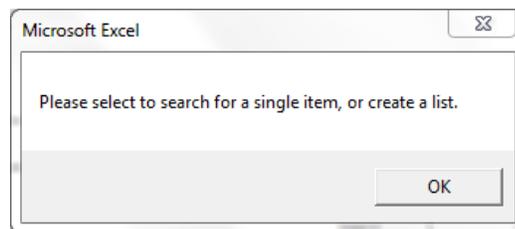


Figure 9 – Search Option Message Box

The search box itself was one of the more difficult features to create. I used a system similar to the one I developed for the USDA assignment in class. Basically, I created a Boolean flag variable for each search criteria. These flags were defaulted as true. If the criteria had a value, the system would check the current row against these criteria to see if the values matched. If they did, the flags remained true. If the criteria didn't match, the flags would be changed to false. If certain criteria were not filled, these flags would remain true. At the end of the routine, I would use an If/Then statement to ensure that all flags were true. If they were, then I would either stop on that row, or add the row to the user defined list depending on the search option.

```
'Check if Vendor matches search
  If cboVendor.value = "" Then
    vcheck = True
  Else
    If Inv.Cells(r, 2).value = cboVendor.value Then
      vcheck = True
    Else
      vcheck = False
    End If
  End If
End If
```

Figure 10 – Sample of search flag code

Inventory Count – The “Inventory Count” feature was also a difficult feature to create. I wanted to have the system search the entire Inventory table and sum total items and total and average value for all items divided up by category, type, and condition. The idea behind this list is to create a simple list that can be used every time the Academy needs to perform an inventory audit. This list will show them everything the system thinks that they have. To accomplish this feature, I needed to have the system divide up inventory items by category, type, and condition before summing. The trouble I had is that each of these three column values is customizable as is noted later on. I needed to be sure that the system would identify and sum any new column values added later on. My solution to this problem may not be the most efficient way of processing the list, but it works and it does not take long to do, so I stuck with it. To generate the inventory count list, the system first goes to the hidden menu table, finds the values for all three columns and prints rows in the InvCount table for every possible combination of category, type, and condition using embedded Do/Loop statements.

```
'Set all possibilities
Do Until Menu.Cells(o, 5).value = ""
  t = 1
  Do Until Menu.Cells(t, 2).value = ""
    a = 1
    Do Until Menu.Cells(a, 3).value = ""
      InvCount.Cells(r, 1).value = Menu.Cells(a, 3).value
      InvCount.Cells(r, 2).value = Menu.Cells(t, 2).value
      InvCount.Cells(r, 3).value = Menu.Cells(o, 5).value
      r = r + 1
      a = a + 1
    Loop
    t = t + 1
  Loop
  o = o + 1
Loop
```

Figure 11 – Setting all possibilities for inventory count

Once these rows are generated, the system uses another Do/Loop statement to go down each row in the InvCount table, find the three column values and then search for them in the Inventory table and sum both the Count and Cost columns where the row criteria matches the criteria in the InvCount table.

```
'search inventory for matches
Dim i As Long
Dim count As Integer
Dim value As Currency
r = 6

Do Until InvCount.Cells(r, 1).value = ""
  count = 0
  value = 0
  i = 2
  Do Until Inv.Cells(i, 1).value = ""
    If InvCount.Cells(r, 1).value = Inv.Cells(i, 4).value Then
      If InvCount.Cells(r, 3).value = Inv.Cells(i, 6).value Then
        If InvCount.Cells(r, 2).value = Inv.Cells(i, 5).value Then
          count = count + Inv.Cells(i, 13).value
          value = value + (Inv.Cells(i, 11).value - Inv.Cells(i, 12).value)
        End If
      End If
    End If
    i = i + 1
  Loop
```

Figure 12 – Search inventory for matches

Once the entire inventory has been counted and summed in the InvCount table, the system looks for rows with zeros in the Count column and deletes those rows. This leaves only rows that have inventory items in them. Finally, the system calculates the average cost per item for each row and inserts that value in its corresponding column and totals the count and cost values and inserts totals at the top of the report.

	A	B	C	D	E	F	G	
1		Total Inventory Count						12/4/2010
2		Utah Valley Fencing Academy, Inc.						
3		Total Count	Total Value					
4		262	\$ 2,291.75					
5	Category	Type	Condition	Count	Average Cost	Total Cost		
6	Barrel	Epee	New	6	\$ 2.23	\$13.39		
7	Base	Epee	New	10	\$ 2.23	\$22.31		
8	Screw	Epee	New	20	\$ 0.25	\$4.90		
9	Socket	Epee	New	7	\$ 3.96	\$27.72		
10	Spring	Epee	New	40	\$ 0.25	\$9.82		
11	Tip	Epee	New	6	\$ 2.98	\$17.85		
12	Wire	Epee	New	1	\$ 2.79	\$2.79		
13	Barrel	Foil	New	9	\$ 2.23	\$20.08		
14	Base	Foil	New	10	\$ 1.86	\$18.59		

Figure 13 – Inventory Count List

Calculate Totals – The Final feature in the Inventory button cluster was created to facilitate the transfer of information from the inventory system to the Academy’s QuickBooks accounting records. The Academy does not keep an item-by-item record of inventory in their accounting records, but instead has three general categories of inventory with their respective values: Weaponry, Attire, and Misc. In the inventory system, each category is assigned to one of these three areas when created. The Calculate Totals button simply opens a form that totals the values for each of these three areas from the inventory table. It is important to note that these values shown are not net of depreciation because the inventory values are stored separately from the depreciation values in QuickBooks.

Inventory Totals		
	Count:	Value:
Weaponry:	219	\$1,691.81
Attire:	23	\$772.55
Misc:	20	\$262.18
Total Inventory:	262	\$2,726.54
<i>Values are NOT net of depreciation</i>		

Figure 14 – Inventory Totals for transfer into QuickBooks

Not Our Goods:

The Not Our Goods button cluster contains all of the tools designed to facilitate tracking of items that are resold to students. As students progress through the fencing courses, they are encouraged to start purchasing their own equipment. To help students get the very best equipment for the best price and to provide additional income for the Academy, UVFA

purchases the items on behalf of the students from their suppliers at a discount and then resells the items to the students at a slightly higher price. A system was needed to track which items purchased belonged to which student, and which students have paid. Additionally, the Academy needed a way to track margins made on purchases to enter the value into QuickBooks.

Add Item - The “Add Item” button in the “Not Our Goods” cluster allows the user to save items that have been purchased for a student and print an invoice to bill the student for the items purchased. This button works similarly to the “Issue Item” button in that it uses a two-dimensional array and a multi-column list box to display the added items. Items can be added to or deleted from the array until the user has everything in the list box that he would like to appear on the invoice. Pressing “Save & Print” then moves the user to the “NOG Invoice” table and uses a Do/Loop statement to fill the rows in the invoice. For the invoice to print correctly on one page, the system has a counter that only allows 34 rows to be saved in each invoice. If a user wants to issue more than 34 items, they will simply issue multiple invoices. This should not be a problem, however, because rarely (if at all) will an invoice meet or exceed 34 items.

Item	Category	Price
Sword	Sabre	52.95
Lame	Sabre	162.50
Glove	N/A	12.95

Date: 12/08/2010
Purchased For: Gove Allen
Category: Jacket
Type: N/A
Size: L
Gender: Male Female
Hand: Right-Hand Left-Hand
Cost: 69.85
Resale: 75.50
Notes: Embroidered "Gove"

Buttons: Delete Item, Save & Print, Add Item

Figure 15 – Add NOG Item User Form

Utah Valley Fencing Academy, Inc.
 728 South State Street
 Orem, UT 84058
 (801) 266-2421
 http://www.fencingutah.org

INVOICE
 Student Purchase

Invoice Date: 12/8/2010

Student: Gove Allen

Line#	Item	Description	Price
1	Sword		\$22.00
2	Lame		\$162.00
3	Glove		\$12.00
4	Jacket	Embroidered "Glove"	75.00
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			

Utah Valley Fencing Academy, Inc.

Figure 16 – NOG Purchase Invoice

Print Statement - The “Print Statement” button was probably the most difficult button to do. The idea behind this button was to provide the user the opportunity to reprint statements showing all unpaid items belonging to a particular user. This can be useful if the student loses an invoice or to view outstanding purchase amounts for any student.

What made this form difficult was that the form needed to contain a list of all students listed in the NOG table. Since many students were listed multiple times, I needed to find a way to eliminate duplicate names. I tried for some time to get a list where only one name appeared, but to no avail. I then searched in Google and found someone that had done something similar in a forum. I was able to modify that person’s code slightly and create a list of individual names in a list box. From there, the form was simple. The system prints the invoice from the “Add Item” button above with any items listed as unpaid that contained the name highlighted in the “Print Statement” form’s list box. The system would know if an item is unpaid if the “Paid On” date column for that row had a null value.

Figure 17 – Print Statement User Form

```
Private Sub UserForm_Initialize()  
    ' This example is based on a tip by J.G. Hussey,  
    ' published in "Visual Basic Programmer's Journal"  
    Dim AllCells As Range, Cell As Range  
    Dim NoDups As New Collection  
    Dim i As Integer, j As Integer  
    Dim Swap1, Swap2, Item  
  
    ' The items are in A1:A105  
    Sheets("NOG").Cells(2, 9).Activate  
    Set AllCells = Range("I2", Selection.End(xlDown))  
  
    ' The next statement ignores the error caused  
    ' by attempting to add a duplicate key to the collection.  
    ' The duplicate is not added - which is just what we want!  
    On Error Resume Next  
    For Each Cell In AllCells  
        NoDups.Add Cell.value, CStr(Cell.value)  
        ' Note: the 2nd argument (key) for the Add method must be a string  
    Next Cell  
  
    ' Resume normal error handling  
    On Error GoTo 0  
  
    ' Sort the collection (optional)  
    For i = 1 To NoDups.Count - 1  
        For j = i + 1 To NoDups.Count  
            If NoDups(i) > NoDups(j) Then  
                Swap1 = NoDups(i)  
                Swap2 = NoDups(j)  
                NoDups.Add Swap1, before:=j  
                NoDups.Add Swap2, before:=i  
                NoDups.Remove i + 1  
                NoDups.Remove j + 1  
            End If  
        Next j  
    Next i  
  
    ' Add the sorted, non-duplicated items to a ListBox  
    For Each Item In NoDups  
        lstNames.AddItem Item  
    Next Item  
End Sub
```

Figure 18 – Code found to delete duplicates

Record Payment – After a student’s items are purchased and delivered, payment is required from the student. The system has a column to record when items are paid for. I decided to record purchases item-by-item, because some students don’t pay for all of the items at once, and some items also don’t arrive at once even though we printed them on the same statement. The record payment pop-up form has the same search features seen in Issue Inventory with the ability to move through the NOG table using previous and next buttons or by selecting a specific row and jumping to it. The form has updating labels that show the data for each row to help the user identify the correct row. When the correct row is found, the user can enter the date the item was paid and press the “Save” button. From then on, when the form passes over that item’s row, the paid date will appear in the “Date of Payment” text box. This date can be edited or deleted by the user by simply including a new value in the text box and pressing “Save.”

Date:	12/8/2010	Student:	Gove Allen
Category:	Glove	Type:	N/A
Size:	10	Gender:	Male
Hand:	Right	Total Cost:	12.95

Go Previous Next

Date of Payment: Save

Figure 19 – Record Payment User Form

Administration:

The Administration button cluster contains tools to edit the lists that populate the drop-down pick boxes as well as tools and calculators to manage depreciation figures and to calculate information to transfer to the Academy's QuickBooks accounts.

Edit Lists – The Edit Lists tool is one of the tools I am most proud of. It began as four different user forms that were all identical each consisting of a list box of items for each list with buttons to add or remove items from the lists and cancel or save the changes. The form was taken from previous in-class activities and basically loaded the list box from the items in the column for that list from the hidden Menus table. When the save button was clicked, the corresponding column was cleared and the list box data was saved over top. In order to reduce the number of forms and corresponding buttons, I learned how to use a multi page tab box where I could include each form on a separate tab and reduce them down to one single form and button.

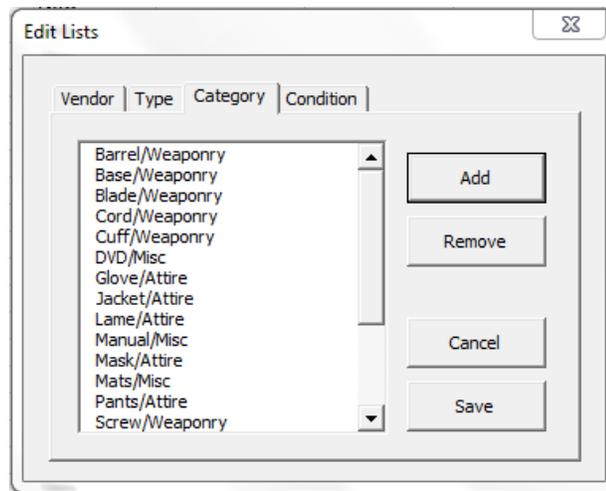


Figure 20 – Edit Lists form with Category columns displayed

Another challenge with this form was that one list box actually needed to contain information from two separate columns. The categories for the items needed to have a corresponding area attached to them in order to make the calculations in the "Calculate Totals" tool work properly. I had to find a way to load both columns into the list box so that each row has its corresponding values and then be able to separate them back into two columns when the save button is selected. My solution was to save the list box values with a concatenation of the two columns separated by a "/" and shown as one value. When items are added to this list, there are two pop-up input boxes (one for each of the columns). These values are also concatenated into one value and added to the list. When the save button is pressed, the values are saved as one value in the first column, and then a text-to-columns command is run to separate the values in the first column using the "/" as a delineator. In this way, the values return to their two columns with each row containing its corresponding values correctly.

```
Private Sub cmdaSave_Click()  
    Menu.Activate  
    Dim r As Integer  
    'set new menu lists for vendor  
    Menu.Range("c:d").Clear  
    For r = 1 To lstCategory.ListCount  
        Menu.Cells(r, 3).value = lstCategory.List(r - 1)  
    Next  
    Columns("C:C").Select  
    Selection.TextToColumns Destination:=Range("C1"), DataType:=xlDelimited, _  
        TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=False, _  
        Semicolon:=False, Comma:=False, Space:=False, Other:=True, OtherChar _  
        :="/", FieldInfo:=Array(Array(1, 1), Array(2, 1)), TrailingMinusNumbers:=True  
    Inv.Activate  
End Sub
```

Figure 21 – Code to paste values into multiple columns

Calculate Depreciation – The depreciation calculation form has three basic functionalities. On the top of the form, it shows the current depreciation values separated into the three QuickBooks accounts mentioned in the “Calculate Totals” tool. These are used to find the current depreciation values and make adjustments to QuickBooks where necessary to match them. The second section is a search box identical to the one used in the “Record Payment” tool and is used to adjust depreciation item by item where necessary. The final section was the most complicated, but mostly because of the calculations involved. Depreciation for this equipment is normally five years straight-line depreciation with no salvage value. This means you simply divide the value of the inventory by five and you have depreciation. The complication comes if this is the first year of depreciation. In this case, the item could be depreciated at half of the year’s value, or by halves of quarters, depending on when they were purchased. To make these calculations possible I left it to the user to determine if items purchased this year would use the half-year or quarter method for calculation, but the system still needed to recognize items purchased in this year and which items were purchased in which quarter to work. I was able to accomplish this by playing around until I figured out how to use relative dates by treating the dates like numbers and using them in calculations. I found I could relate them using < or > and add or subtract days with them to get relative dates. Using calculations, I was able to get the system to determine quarters and half-year calculations based on a year-end date input by the user. I wanted to hard code this date at first, but the Academy may consider moving its year-end to June, so I decided to leave the year-end date for depreciation calculation as a user-input value in the event it should change in the future.

```
r = 2
Do Until Inv.Cells(r, 1).value = ""
  InvDate = Inv.Cells(r, 1).value
  InvAmt = Inv.Cells(r, 11).value

  If InvDate < (Now() - 365) Then
    DepAmt = (InvAmt / 5)
  Else
    If btnHalf.value = True Then
      DepAmt = (InvAmt / 10)
    End If
    If btnQuarter.value = True Then
      If InvDate < (InvDate2 - 274) Then
        '1st Quarter
        DepAmt = (((InvAmt / 5) / 4) * 0.5)
      Else
        If InvDate < (InvDate2 - 183) Then
          '2nd Quarter
          DepAmt = (((InvAmt / 5) / 4) * 1.5)
        Else
          If InvDate < (InvDate2 - 92) Then
            '3rd Quarter
            DepAmt = (((InvAmt / 5) / 4) * 2.5)
          Else
            '4th Quarter
            DepAmt = (((InvAmt / 5) / 4) * 3.5)
          End If
        End If
      End If
    End If
  End If
End If
```

Figure 22 – Using dates in calculations

Depreciation

Weaponry Depreciation: \$ 270.5
Attire Depreciation: \$ 126.88
Misc Depreciation: \$ 37.42

Edit Single Depreciation Amount

Category: Spring Type: Foil
Date: 9/13/2010 Condition: New
Count: 10 Total Cost: 2.805

Go Previous Next

Depreciation Amount: 0.35 Save

Add Depreciation

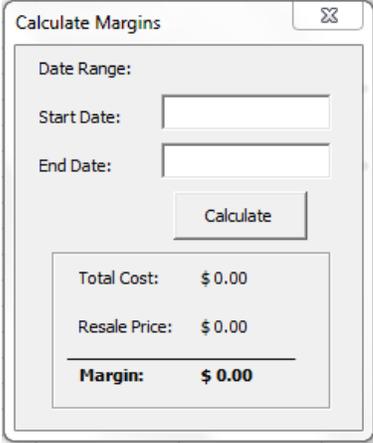
Add Depreciation as of: 01/01/2011

1/2 year Quarters Add

Figure 23 – Depreciation Calculator Form

Calculate Margins – The final feature in the inventory system is the margins calculator. For every item sold to students through the NOG section, there is a small margin that the Academy makes. These margins will need to be entered as earnings into QuickBooks periodically. To facilitate this process, the calculator will take a start and end date from the user to create a range to calculate

the margins for. The system will then run a simple Do/Loop command to tally the price, cost, and margin for all items within that date range, and then display those figures using labels on the form.



The image shows a dialog box titled "Calculate Margins" with a close button in the top right corner. Inside the dialog, there is a section labeled "Date Range:" containing two input fields: "Start Date:" and "End Date:". Below these fields is a "Calculate" button. At the bottom of the dialog, there is a summary table with the following data:

Total Cost:	\$ 0.00
Resale Price:	\$ 0.00
Margin:	\$ 0.00

Figure 17 – Margins calculator for NOG purchases

Although many tools in this system like the calculators are very simplistic in nature, they cut down so much manual calculation and greatly facilitate the accounting efforts at the Academy. The fact that this system was completely customized to the Academy and their accounting records has made its potential even more valuable. The Academy – as well as myself – is very excited to begin implementing this system in their Armory and I am sure new improvements and further suggestions will arise as we begin full implementation.

