Joseph Gridley

VBA Project

4-13-11

Executive Summary

The Problem

The Applewood Estates Homeowners Association treasurer is in need of a new invoice system. His responsibilities include invoicing each member of the HOA each year for the HOA fees. The software they had been using is antiquated and generally ill-suited for the needed functionality. They had been using a purchase order form to create the invoices. The computer with the software is on its last legs creating further urgency in finding a replacement solution.

The HOA mails invoices once a year for fees as well as subsequent months for delinquent accounts. No other invoices are sent. A simple interface for quickly creating invoices is needed.

The Solution

The treasurer is proficient in Excel and came up with the idea and functionality of the Excel implementation. The focus is on the speed of creating invoices. The solution includes the following features:

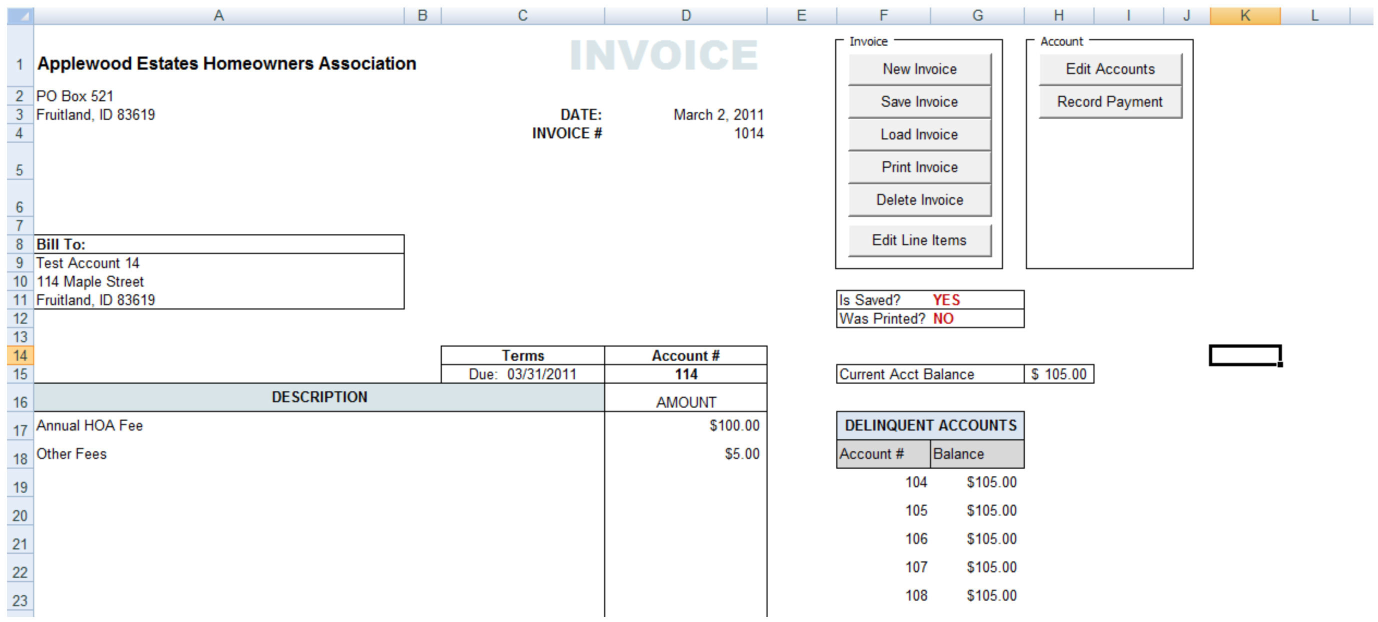
- Historical data retention
- Create, Read, Update and Delete invoices, accounts, and line items to be billed.
- Each feature can be done at least two different ways through the use of forms as well as controls directly in the worksheet.
- The sheet will indicate if the invoice has been saved or not as well as if it has been printed or not.
- Payments can be entered through a form and the account balance will be kept.
- Delinquent accounts will show up based on the current date and the invoice due dates.

Implementation Documentation

Everything that needs to be done can be done through one worksheet. Other worksheets are hidden but could also be used to enter in new data. These alternate forms of entering data will be discussed later.

Initial Overview

As you can see below, the invoice that will be printed is created directly on the sheet. When opening the workbook, the Date in cell D4 will be updated to the current date. The 'Bill To:' section is populated based on the account number in D15. The line-items in the description section are dropdown boxes with predefined line-items and the amount changes to the default amount when selected. However, the amount can be changed manually.



Applewood Estates Homeowners Association
PO Box 521
Fruitland, ID 83619

INVOICE

DATE: March 2, 2011
INVOICE # 1014

Bill To:
Test Account 14
114 Maple Street
Fruitland, ID 83619

Terms	Account #
Due: 03/31/2011	**114**

DESCRIPTION	AMOUNT
Annual HOA Fee	$100.00
Other Fees	$5.00

Invoice: New Invoice, Save Invoice, Load Invoice, Print Invoice, Delete Invoice, Edit Line Items

Account: Edit Accounts, Record Payment

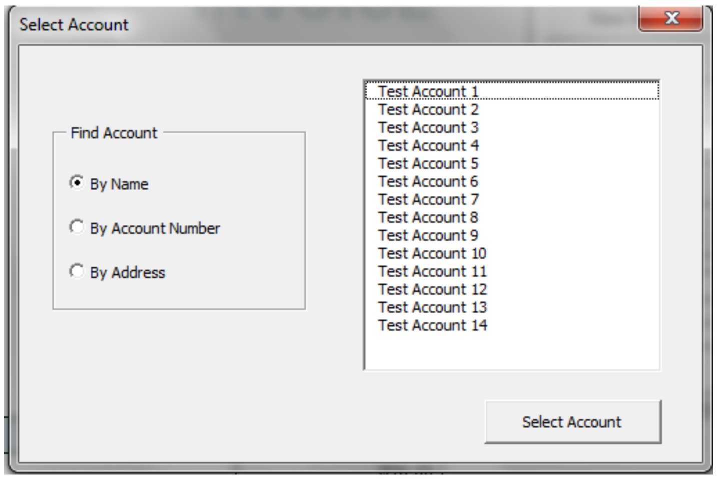
Is Saved?	YES
Was Printed?	NO

Current Acct Balance	$ 105.00

DELINQUENT ACCOUNTS

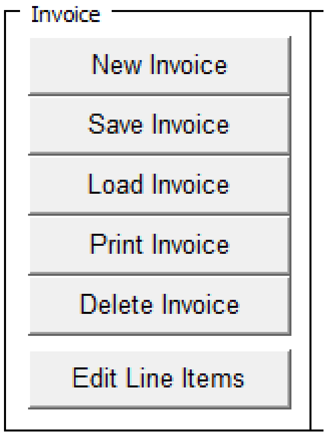
Account #	Balance
104	$105.00
105	$105.00
106	$105.00
107	$105.00
108	$105.00

The account number in D15 can be selected through the dropdown box in that cell or the cell can be double-clicked to bring up this form:



Through this form, an account can be selected by choosing the name, account number or address. The user can double-click the desired account or highlight the desired account and click select account. The form will then close and D15 will have been set to the correct account number.

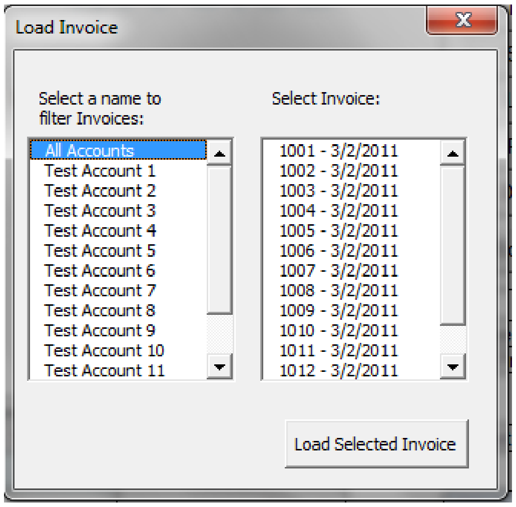
Now I will discuss the invoice button section on the right as shown below:



New Invoice: The 'New Invoice' button increments the invoice number, resets the date, and clears the account number. The rest of the previous invoice is left the same as each invoice, for the most part, will be identical (user's request).

Save Invoice: This button saves the current invoice data onto another spreadsheet used to store historical invoice data. When saved, the 'Is Saved' value is changed to 'YES'. Each time you change a value on the invoice, a subroutine runs to see if the values on the invoice match what has been saved. If they match, the value changes to 'YES'; if they don't, the value changes to 'NO'.

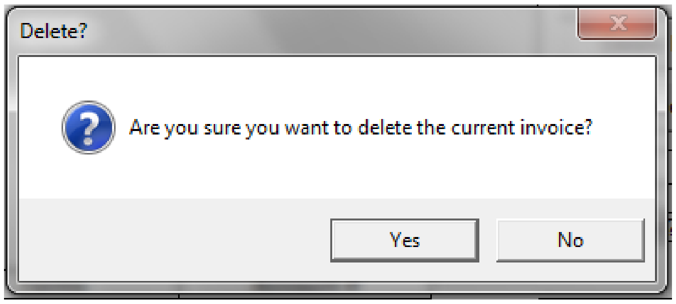
Load Invoice: This button opens the 'Load Invoice' form, which allows you to open any previous invoice. As you can see in the figure below, you can filter the invoices by selecting a specific account name on the left. When 'All Accounts' is selected, all invoices will show on the right.



Select a name to filter Invoices:	Select Invoice:
All Accounts	1001 - 3/2/2011
Test Account 1	1002 - 3/2/2011
Test Account 2	1003 - 3/2/2011
Test Account 3	1004 - 3/2/2011
Test Account 4	1005 - 3/2/2011
Test Account 5	1006 - 3/2/2011
Test Account 6	1007 - 3/2/2011
Test Account 7	1008 - 3/2/2011
Test Account 8	1009 - 3/2/2011
Test Account 9	1010 - 3/2/2011
Test Account 10	1011 - 3/2/2011
Test Account 11	1012 - 3/2/2011

Print Invoice: This button pulls up the print dialog box and allows you to print the invoice. After printing, the 'Was Printed' value for that invoice will be changed to 'YES'. This data point is kept on the invoiceDB sheet and a formula pulls the yes or no from that sheet based on the invoice number.

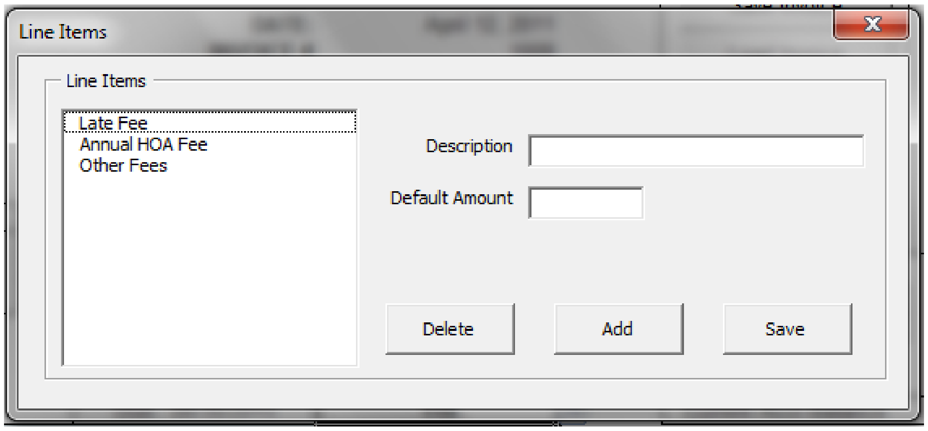
Delete Invoice: When the 'Delete Invoice' button is clicked, the following dialog box appears:



Are you sure you want to delete the current invoice?

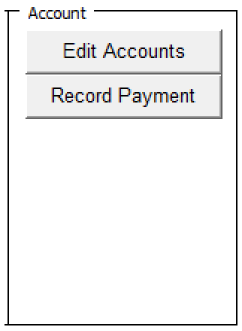
If the user clicks 'Yes', the current invoice's data will be removed from the invoiceDB sheet and the 'New Invoice' subroutine will run.

Edit Line Items: This button will open the following form:

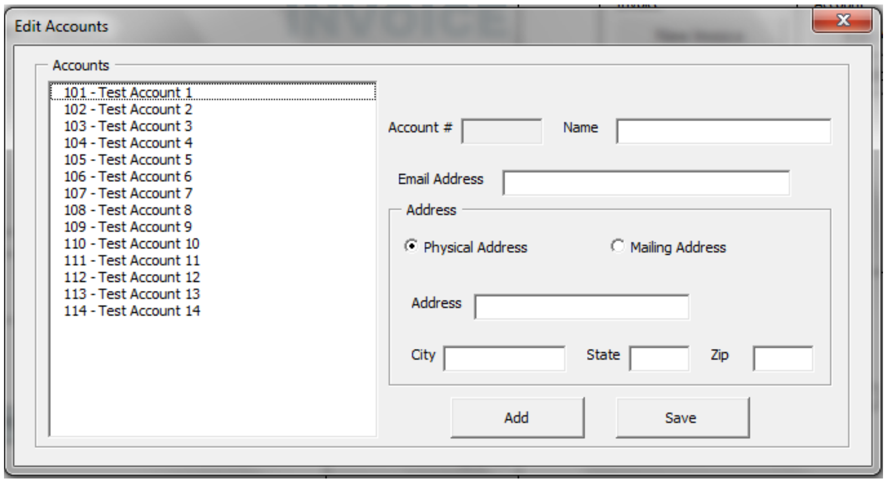


The user can click the line items on the left in order to edit or delete the line item. If the user would like to add a new line item, it can be done by simply editing the description, editing the default amount, and clicking add.

The next button section looks like this:

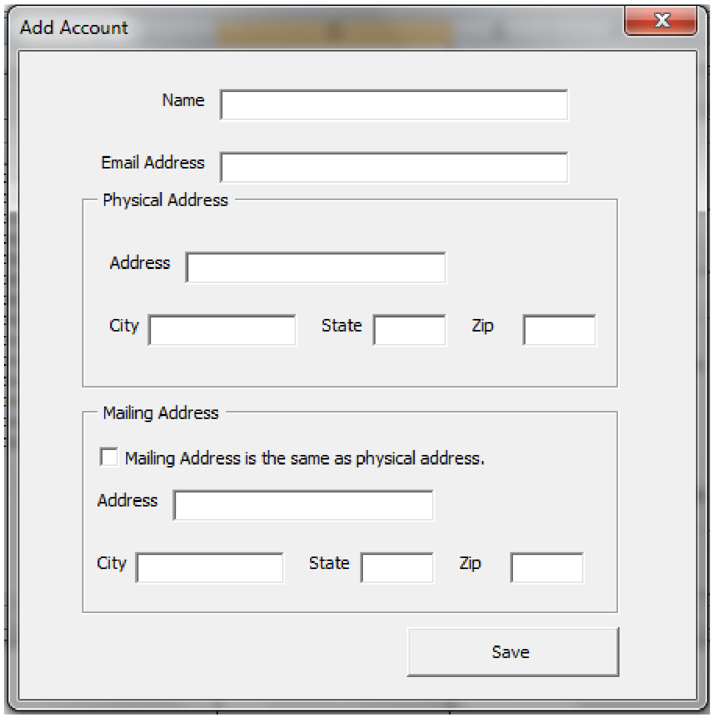


Edit Accounts: This button brings up this form:



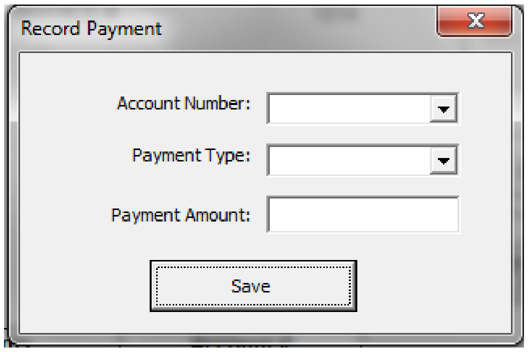
101 - Test Account 1
102 - Test Account 2
103 - Test Account 3
104 - Test Account 4
105 - Test Account 5
106 - Test Account 6
107 - Test Account 7
108 - Test Account 8
109 - Test Account 9
110 - Test Account 10
111 - Test Account 11
112 - Test Account 12
113 - Test Account 13
114 - Test Account 14

The accounts are listed on the left in the format of <account number> - <account name>. The information can be edited on the right. The user can add a new account by clicking the 'Add' button which will bring up another form that looks like this:



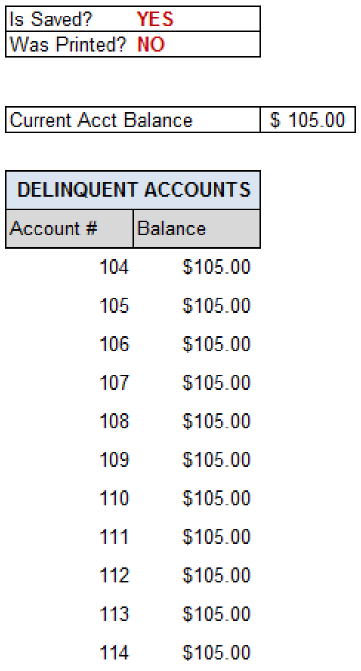
This form allows the entry for the details of a new account. Clicking the checkbox to indicate that the mailing address is the same as the physical address will fill in the mailing address from what was entered into the physical address section.

Record Payment: This will bring up this form:



The account number drop-down box in auto-populated with all account numbers. The payment type can be "Check", "Cash", or "Credit Card". The payment amount is entered and the user presses the save button. The payment is recorded and the delinquent account list(explained below) is refreshed in order to reflect any changes after the payment is recorded.

Below is a screenshot of a portion of the worksheet:



Is Saved?	YES
Was Printed?	NO

Current Acct Balance	$ 105.00

DELINQUENT ACCOUNTS	
Account #	Balance
104	$105.00
105	$105.00
106	$105.00
107	$105.00
108	$105.00
109	$105.00
110	$105.00
111	$105.00
112	$105.00
113	$105.00
114	$105.00

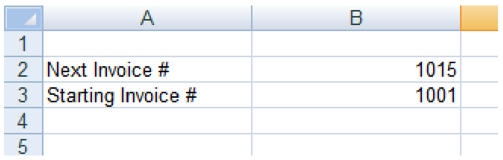
"Is Saved" and "Was Printed" have been discussed previously but are shown here to orient the user as to where this is located on the sheet. The current account balance is the balance of the currently selected account number on the invoice.

'Delinquent Accounts' is a list of accounts that didn't pay their balance by the due date which is the last day of the month that the invoice was generated in (if you are using this spreadsheet for a different project, this would most likely need to be changed). This list is updated as soon as you open the spreadsheet as well as when a payment is recorded.

Here is a quick overview of the other sheets:

DATABASE: This sheet houses the account information. Accounts can be added here without using the form and the named ranges will automatically update. The named ranges are used to populate the dropdown boxes on the invoice.

Admin: This contains a formula that finds what the next invoice number. If no invoices exist, the next invoice number will be what is entered as the starting invoice number as seen in the screenshot:



	A	B
1		
2	Next Invoice #	1015
3	Starting Invoice #	1001
4		
5		

This sheet also houses the line item data. The line items can be edited directly on this sheet and named ranges will automatically update.

InvoiceDB: This sheet contains the invoice data. It is not recommended to update this sheet directly because balances wouldn't update. However, a user could take advantage of the layout to make many invoices very quickly, understanding that the balances won't update until a payment is made or the invoice is loaded and saved through the buttons on the main invoice page.

Balance: This holds the current balance of each account. This is updated whenever an invoice is saved or a payment is made.

Payments: This holds the payment information. It is not recommended to updated this sheet directly as the balance will not update. Payments should be entered through the "Record Payment" button.

Learning and Conceptual Difficulties

This project was a great opportunity to take advantage of the events available in VBA. I used many events that I hadn't ever used before. For example, the workbook_open event was used to update the date and the delinquent account list. The worksheet_change event was used to update named ranges and to check if the invoice was saved or not.

One difficult concept was how to know if the invoice was printed or not. There is a BeforePrint event but not an AfterPrint event. I ended up not using an event but wrote a subroutine that opens the print dialog box which returns true or false depending on what button was ultimately pressed. I then assigned ctrl + p as the shortcut key so this subroutine would run and the subroutine would catch whether the user actually printed or not. In the end, if the user selects print through the traditional menu, the spreadsheet will not record the invoice as having been printed.