|  |
| --- |
| Simply Invoicing |
| Spreadsheet Automation Final Project |
| ShaLae Steadman |

# Executive Summary of Simply Invoicing

Simply Invoicing is a project designed specifically to help assist the functions of Steadman Sound Inc.

**Background of Project:** My father is an extremely hard working business owner who does not like to paperwork. This project is to allow him to quickly complete and email his invoices on the job.

**Purpose of Project:** to create an easy to use method for creating, saving, and emailing invoices from Steadman Sound Inc. The entire process should be completed in minutes.

**Overview of the Project:** This project uses a custom tab to open and input data. It uses two forms that are most automatically filled out. The forms are then documented, saved as pdfs, and emailed out to clients as attachments. The information can be saved for a later email date.

# Project Background

My father, Glen Steadman, is an incredibly hard worker. He has created and built up his own business, Steadman Sound Inc., specializing in sound. He contracts out to primarily to facility management groups with the Church of Jesus Christ of Latter-day Saints. He travels throughout Utah and even provides services in other boarding states.

My father loves his job. He loves his company. He is great at working with different people and finding solutions to sound problems. He also loves the flexibility of his job. However, my father does not like doing “paperwork.” He works very hard during the day and when he comes home he would like to be able to relax and spend time with his family. Therefore, often times invoicing for the jobs he has completed goes undone. Many times it isn’t until my mom reminds him that there has been any money going into their account, does he take the time to send out his invoices.

In order, to make it easier for my father to complete his invoices on the job, I have created this simple VBA project to automate the process. Now, once my father has completed his job, he can sit down immediately and send out the invoice on his laptop. This will allow him to complete his job at work and be able to focus on his family at home.

# Overview of the Project

This VBA project creates and emails invoices to clients for Steadman Sound Inc. The invoice form is completed automated so that creating and emailing an invoice takes a matter of minutes. The data is saved and stored for easy use in the future. Please note that the data shown in this spreadsheet is not real data. It has been created for purpose of presentation and to maintain privacy. Steadman Inc. will use its own client information.

# Using the Project

In order for the user to create an invoice he will open the spreadsheet and click on the custom tab titled Invoices. 

## Adding a New Invoice

Here, the user will be able to add a new invoice or email a previously inputted invoice. If the user clicks on the “Add New Invoice” button, the following form will open.



As shown, the current date is automatically filled in along with a new invoice number. The user than will be able to select the customer from a drop down combo box. Once the customer is selected, the customer’s email, address, terms and beginning of the PO Number will be automatically filled in the respected spaces. This data is being pulled from the “Customer” tab of the workbook. The user will need to input the last three digits of the PO Number and the Building number as these change with each new invoice. This data will be given from the customer.

The next step is to choose an item(s), or service(s) that was performed for the customer. The user can select up to five items from the drop down list. Once these items are selected, their account, description, amount charged are automatically filled in. This data is pulled from the “Items” tab of the workbook. The code used to allow for the automation is show to the below. 

Once all of the items are recorded, the total owed is calculated. As most clients of Steadman Sound Inc. are tax exempt, taxes are automatically set at zero. However, the user may change the tax category and the total will automatically be updated to reflect the additional cost.

At this point the entire form is completed. However, the user may manually update any input box for their own purposes. Once the user has completed the invoice, they can do the following: (1) save and add a new record, save the information and exit the form, email the invoice to the client, or cancel and exit the form without saving the information.

### Saving the Information

The information inputted on the form will be saved in the “Invoice” tab of the workbook. Each new record will be saved on row 2. That way the user can easily see the most recent record. If the user does not email the invoice before saving it, it will be recoded in column “B” with a no.

### Adding a New Record

If the user decides to enter a new record, the current form will save and close. Then a new form will automatically open with a new invoice number.

### Emailing an Invoice

If the user decides to email the invoice, the project will determine if all needed information is recorded. For example, if the user does not have the last three digits of the PO number, a message box will pop up asking for the information. The invoice will not be able to be sent out with a PO number and Building Number.

 Once all required data has been recorded, the user will know that the email is sent by receiving a message. The information will be saved as described above with the emailed checked.

In order, for the user to receive the “Email Sent” message, the project completes the following steps:

1. It inputs the data into the “Invoice Template.



1. The template is automatically reconfigured to fit all of the text.
2. The template is saved as a pdf. The file is saved in the folder “Invoices” using the Invoice Number as the identifier.



1. The recently save PDF is then attached to an email and sent to the client.

### Canceling the Form

If the user cancels the form, the form will close out and no data will be saved. However, the user will be given a warning an option to go back and save the data before it is lost.

## Emailing a Recorded Invoice

If the user wants to go back and quickly send all unsent invoices, the user only needs to click on the second button of the custom tab. This button will pull up a very similar user form with the recorded information of an invoice that has not been sent. However, if all recorded invoices have been sent the user will receive a message. 

The user will be able to change or edit the input. Then will have following five options with the form: (1) create a new invoice, cancel the form, email the invoice, save and close the form, or view the next non emailed record. The new, cancel, email invoice, and save and close buttons do the same functions as listed above.

### Viewing the next record

The user can see other un-emailed records by clicking the next button. This simply pulls in the data from the “Invoice” Tab. The user can review each unsent invoice in this matter. Any edited data is recorded back in to the “Invoice” Tab.

Therefore, this project is very simply used by any user. They will be able to quickly send and review invoices.

# Lessons Learned from the Project

I learned a lot of lessons on this project. First, I learned that you cannot unload a form during an initializing it. I was trying to get the form to close if there were no additional invoices to view, however this is not possible. Second, I really learned how to perfect my skills with creating and using user forms. I not only learned how to edit the font and size of text, but I also learned how to powerfully use combo boxes.

Third, I learned how to email using VBA. I have never automated email before and I learned how simple it was to do. However, it took me while to figure out the correct code for saving a pdf with the title changing for each new save.

Forth, I feel that I have become proficient with For/Next Loops, If statements, and With clauses. I used these tools a lot to automate my forms.

In general, I really enjoyed working on this project. I liked knowing that I was solving a real life problem and that my work was going to be implemented. However, I did encounter so frustrations in creating this project. For example, it took me a long while to figure out how to sum up the amounts to equal the total. Over and over again I received an error sign. The first issue was that the cells I was trying sum were defined as integers. So I tried to make it read as an integer by summing the cell to zero. However, this still left me with an error term. In the end, with the help of Dr. Gove Allen it was determined that it wouldn’t work because the amount cells in the user form were empty until an item was chosen. Therefore, we created our own formula to bypass this problem.

# Assistance sought for the Project

I received assistance from three people.

1. Kendall Morrison helped me understand the following:

 For x = 0 To Cbcustomer.ListCount - 1

 If Cbcustomer.ListIndex = x Then

1. The Class TA quickly emailed how to input email information.
2. Dr. Gove Allen helped explain that you can close and initialize user forms at the same time. He also provided the following code:

 Function cNum(S As String) As Single

 If S = "" Then

 cNum = 0

 ElseIf IsNumeric(S) Then

 cNum = CSng(S)

 Else

 cNum = 0

 End If

End Function

Overall, I received about 15 minutes of help on this project.