

EXECUTIVE SUMMARY

PROBLEM:

Currently I am a home teaching supervisor in my Elders Quorum in my ward. I am in charge of gathering the reports for my district on whether or not they completed their home teaching as well as if there are any other concerns the Elders Quorum Presidency should know about such as if the family is struggling. As I was trying to fulfill my responsibility I found that there was no good way to group all of the contact information to make it easy to obtain these reports. The only thing I was given was a pdf of who my district members were, their companions, and which families were being taught. There was also no way to digitally enter in home teaching reports, they were to be printed out and filled out manually.

SOLUTION:

My solution I have implemented is a tool designed to create groups and companionships to organize the contact data in a way that is easy to view. Along with this, I wanted the tool to be relatively easy to change group information. For this reason, I worked specifically with the format that is used by the LDS church for a ward list. The tool is designed so that it can be updated by a new ward list without erasing all the data. It also allows data about which families were taught to be retained even if groups or companionships change or if the ward list changes.

I also wanted a way to contact each group in a mass email both using this tool as well as creating a google import list to create the group on google contacts so that a person can contact the entire group on the fly. In addition to making it easier to contact the Elders, I wanted to make it easier to create a report where I could fill out home teaching results.

SYSTEM OVERVIEW:

The system is essentially a list of worksheets that each act as their own database. First a ward list from the LDS Church is imported. The crucial data from the ward list is copied into the tool and acts as a repository for the rest of the data. With the data in place, the tool allows the user to create groups within individuals from the ward list. Once there are groups, the user can edit group memberships. Companionships can be added to groups and are made up of two individuals and the families they teach. Companionships' families can be edited. Both groups and companionships can be edited and deleted. The system also allows for mass emails to whatever group the user chooses. Once the groups and companionships have been formed, the user can then generate a report where home teaching can be recorded. Lastly, I included a function that transfers the data captured in the report and saves it to another worksheet so that data can be preserved even if groups or companionships change.

Below is the ribbon that outlines the main features of my system.



HOME TEACHING REPORTING TOOL

IMPLEMENTATION DOCUMENTATION:

The following section will walk through each of the buttons on the custom ribbon below. It will describe what the function does and how to use it if a form is required. Otherwise all that is required is click of the button on the ribbon.



IMPORT WARD LIST

This button when clicked opens a file dialog. The user should choose a ward list that can be downloaded directly from the LDS Church's system. The procedure opens the file, copies some, but not all of the data and transfers it to the "Ward List" sheet. If the sheet exists already, it is deleted and created again.

EXPORT TO GOOGLE

This button when pressed brings up the following form.

The user simply chooses a group and the procedure iterates through the group and ward list. It takes the values and puts them into a new workbook based off of the Google Template. It then saves the workbook as a csv with the district name in the file name. It is saved in the same folder as the tool. This file can then be imported into Google Contacts. Google contacts automatically puts the contacts into a group with the same name. This allows users to contact the groups in mass email using Google.

ADD GROUP

This button when pressed bring us the following form.

Create a Group

Group Name:
District 4

Create

Close

Vega, Jensen
Savage, Hugo
Brennan, Rodolfo
Mcdowell, Joel
Elliott, Jaxon
Morton, Aljah
Weeks, Enrique
Curtis, Jakob
Morgan, Gilbert
Whitaker, Makai
Barber, Edward
Rocha, Terry
Frazier, Cannon
Espinoza, Ahmad
Mack, Israel
Stevens, Dominik
Rivers, Derrick
Vincent, Kody
Schneider, Barrett
Pollard, Zachery
Daniels, Tobias
Soto, Wesley
Giles, Jamal
Steele, Shane
Murillo, Jude
Fletcher, Leonard
Short, Jonathon
Wallace, Devan
Cervantes, Antonio

The form is initialized by iterating through the ward list to populate the list box. The user then chooses the desired individuals, chooses a name and clicks “Create.” Once create is clicked, a new spreadsheet is made with the same name as the group and all members with their information is copied to it.

EDIT GROUP

This button when pressed brings up the following form.

The 'Edit Groups' dialog box contains the following elements:

- Members List:** A scrollable list of names, including Vega, Jensen; Savage, Hugo; Brennan, Rodolfo; Mcdowell, Joel; Elliott, Jaxon; Morton, Alijah; Weeks, Enrique; Curtis, Jakob; Morgan, Gilbert; Whitaker, Makai; Barber, Edward; Rocha, Terry; Frazier, Cannon; Espinoza, Ahmad; Mack, Israel; Stevens, Dominik; Rivers, Derrick; Vincent, Kody; Schneider, Barrett; Pollard, Zachery; Daniels, Tobias; Soto, Wesley; Giles, Jamal; Steele, Shane; Murillo, Jude; Fletcher, Leonard; Short, Jonathon; Wallace, Devan; Cervantes, Antonio; and Joyce, Bruno.
- Group:** A dropdown menu currently showing 'District 4'.
- Buttons:** 'Delete Group', 'Save', and 'Close'.

This form allows the user to see what members are currently in the group by iterating through the group spreadsheet. It selects the members that are in the group. A user can select or unselect any member and the group will be changed as soon as the user clicks save. It also refreshes the group information by again pulling it from the ward list. The user can also delete the group by clicking “delete group.” This removes the group completely from the workbook, but does not touch the ward list. This also removes all companionships from the “companionships” worksheet.

ADD COMPANIONSHIP

This button brings up the following form.

Create a Companionship [X]

Families

Group :

Companion 1 :

Companion 2 :

Morton, Alijah
Morgan, Gilbert
Frazier, Cannon
Vincent, Kody
Steele, Shane
Nelson, Cortez
Ramos, Vaughn
Davila, Connor

Vega, Jensen & Callie
Savage, Hugo & Donna
Brennan, Rodolfo & Halle
Mcdowell, Joel & Janelle
Elliott, Jaxon & Leanna
Morton, Alijah & Lara
Weeks, Enrique & Elliana
Curtis, Jakob & Tatiana
Morgan, Gilbert & Abigail
Whitaker, Makai & Alondra
Barber, Edward & Dana
Rocha, Terry & Kaley
Frazier, Cannon & Mckayla
Espinoza, Ahmad & Kassandra
Mack, Israel & Scarlett
Stevens, Dominik & Katie
Rivers, Derrick & Deanna
Vincent, Kody & Jayleen
Schneider, Barrett & Kayleigh
Pollard, Zachery & Jessie
Daniels, Tobias & Celia
Soto, Wesley & Yazmin
Giles, Jamal & Daphne
Steele, Shane & Alannah
Murillo, Jude & Lesly
Fletcher, Leonard & Anika
Short, Jonathon & Presley
Wallace, Devan & Danielle
Cervantes, Antonio & Samantha
Joyce, Bruno & Kathy
Guzman, Jordon & Annabelle

Cancel Save

This form allows a user to create a new companionship. The available companions are drawn from the group that is selected. Once the companionship is chosen, the user can then choose the families that are taught. When the user clicks the save button, the companionship is added to the "Companionship" where all the information, including the group is saved.

EDIT COMPANIONSHIP

When this button is pressed the following form appears.

Edit Companionships [X]

Families

Group: District 1

Companionship: Mcdowell, Joel & Morton, Alijah

Delete Companionship

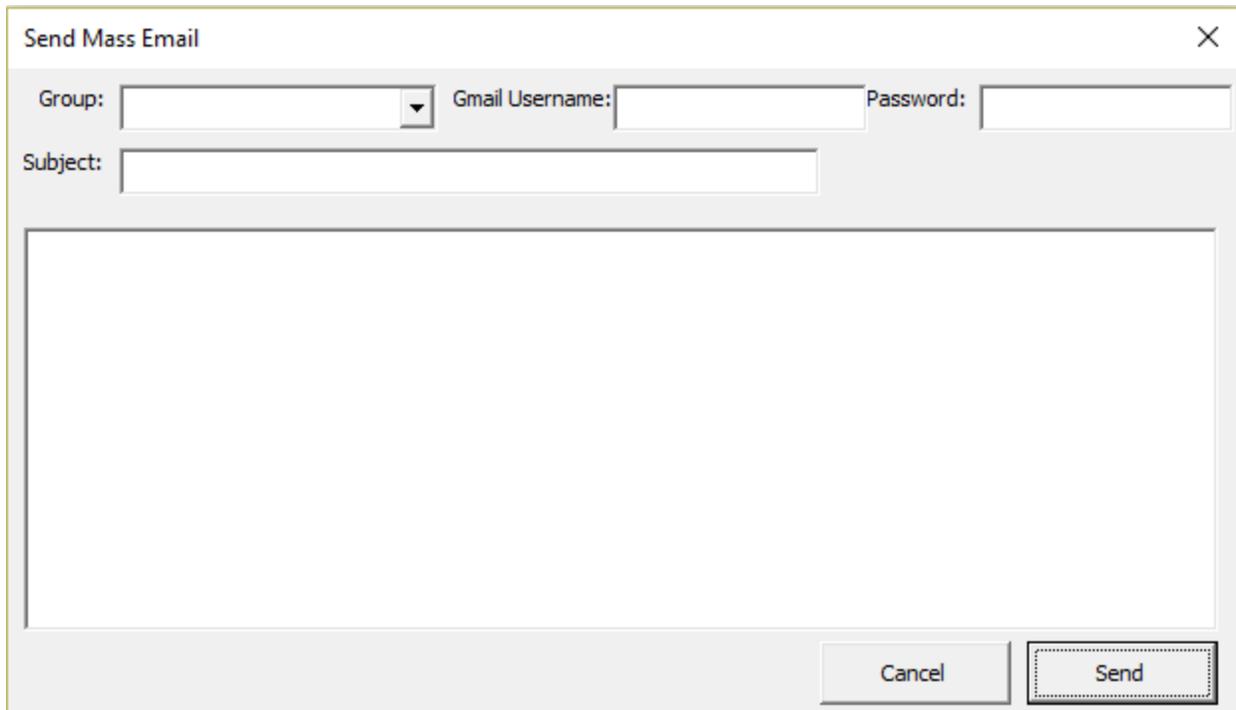
Save Close

Vega, Jensen & Callie
Savage, Hugo & Donna
Brennan, Rodolfo & Halle
Mcdowell, Joel & Janelle
Elliott, Jaxon & Leanna
Morton, Alijah & Lara
Weeks, Enrique & Elliana
Curtis, Jakob & Tatiana
Morgan, Gilbert & Abigail
Whitaker, Makai & Alondra
Barber, Edward & Dana
Rocha, Terry & Kaley
Frazier, Cannon & Mckayla
Espinoza, Ahmad & Kassar
Mack, Israel & Scarlett
Stevens, Dominik & Katie
Rivers, Derrick & Deanna
Vincent, Kody & Jayleen
Schneider, Barrett & Kayle
Pollard, Zachery & Jessie
Daniels, Tobias & Celia
Soto, Wesley & Yazmin
Giles, Jamal & Daphne
Steele, Shane & Alannah
Murillo, Jude & Lesly
Fletcher, Leonard & Anika
Short, Jonathon & Presley
Wallace, Devan & Danielle
Cervantes, Antonio & Sam
Joyce, Bruno & Kathy
Guzman, Jordon & Annab
Woodard, Elisha & Dalia
Houston, Allan & Cora
Rush, Demarion & Aryana

This form allows the user to change the families that each companionship teaches. The companionships are pulled from the companionship list according to which group is selected. If the group changes then so do the available companionships. If the user hits save, the changes are updated in the “companionship” worksheet. If the user clicks “Delete Companionship”, the companionship is deleted from the worksheet.

SEND MASS MAIL

This button generates the following form.

A screenshot of a 'Send Mass Email' dialog box. The dialog has a title bar with a close button (X) in the top right corner. Below the title bar, there are three input fields: 'Group:' with a dropdown arrow, 'Gmail Username:' with a text box, and 'Password:' with a text box. Below these is a 'Subject:' label followed by a text box. A large, empty text area for the email body is positioned below the subject field. At the bottom right of the dialog, there are two buttons: 'Cancel' and 'Send'.

This procedure allows the user to send a mass email to whichever group. I decided that these types of emails did not need to be modified based on the user, but a generic message can be sent to all members. The user needs to choose a group, put in a Gmail email and password. I did have to change my google settings to allow this type of function. The procedure takes the subject and large text field and puts it as the message. I used the SendGmail function designed by Dr. Allen. I also added the ability to have multiple lines on the message text box. The user simply needs to hit Ctrl+Enter to make a new line.

Once the user hits send, the procedure iterates through the emails of the group list. It tries the first email and then the second. It also notifies the user if the email was sent. The user just needs to hit cancel to close the window.

GENERATE REPORT

This procedure iterates through the companionship sheet and creates another worksheet called "Report." It organizes the data in such a way that is easily sorted. It also provides a location to record if home teaching was accomplished. It also creates a column that calculates the percentage of home teaching that has been completed. Clicking this button will overwrite the "Report" spreadsheet if it exists. This button should only be clicked after the update families taught button has been clicked and changes have been made in the groups, companionships, or assignments.

UPDATE FAMILIES TAUGHT

This procedure first checks the ward list to see if there are any new families in the ward list. If there are then they are added to this list. Next, the procedure checks the "Report" spreadsheet to see if there are any documented visits. If there are, it copies them over to "Family Tracking." The purpose of this function and worksheet is to preserve a record of families being taught. This is important to note because, even though companionships change, some families remain in the ward and it is important to track how much they have been visited.

LEARNING:

I already had some experience with VBA before this class, but throughout this class and particularly this project I learned a lot and perfected my skills. Each sub procedure I created had at least one element that I was not super familiar with and had something I had to struggle through. One of the greatest struggles I had was getting the algorithms just right for the edit group and edit companionship forms as well as how to save the home teaching data independent of the groups or companionships. There are some nested loops that can be confusing to comprehend. Understanding the logic was one of my greatest challenges.

The way I handled dealing with the algorithms was by going into break mode a lot and stepping through the program. I felt like this project helped expand my ability to use break mode to trouble shoot problems. In addition to this I learned many functions from using the record macro functionality in excel. I was able to adapt the macros to fit my specific needs.

Another thing I learned was how to better interface with the forms objects. Most of my project is just macros attached to forms and adjusting the form objects as the user works with it. For example, I had to be creative when making the edit companionship form because I wanted the form to dynamically change depending on what the user chose. I had to use other event handlers like change and after_update to handle these changes.

Creating an export that google would recognize was also a learning experience. I knew that it was possible to import contacts into google, but I had to figure out the correct format that would work. I experimented with this and ultimately exported a google contacts csv to view the right format. I also had to reconcile what fields were necessary and what format each field needed to be in. I eventually learned the correct format and how to export the file as a csv. I also used what I learned in the folder picker worksheet how to save the worksheet in the current folder with a dynamically created name.

I also learned the importance of having appropriate scope. Initially my proposal included way too many features than I had time to implement. For example, I wanted to interface with Google Docs, but realized that there is not much value in this for now. My quorum does not use Google Docs and the maintenance required for it is too much. I wanted create a solution that was simple to use and easily adjusted. For this reason, I created a "refresh groups" sub procedure.

Most of this project increased my level of comfort with VBA. I feel like I was able to apply many principles from the assignments and projects we did in class and further solidify them for me. For example, in the ISBN validator we learned about how to manipulate text. This project required me to manipulate the text to work in the forms as well as in the google export. Obviously my learning from the User Forms assignment was crucial in this project. I also implemented concepts from "Choose a Folder" and "Parse Email Address." Lastly, I implemented basic concepts such as do until and for loops as well as if statements.

ASSISTANCE

I did not receive any major assistance from another person. The only assistance I received was utilizing the sendGmail function that was from the "Send Reminders" workbook and a small snippet of code for opening a file that I found online. These code snippets are marked in the comments of the code.