

Final Project Write-Up

Executive Summary

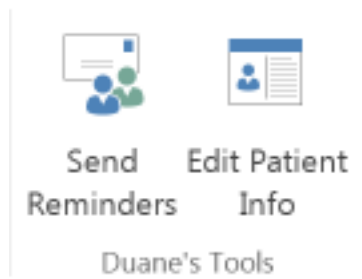
My father-in-law is a chiropractor and runs his own practice. He is looking for a way to send appointment reminders to his patients. My project informs the user of who has an appointment that day and allows the user to send an email or text reminder to those individuals. My project also allows the user to add a new patient or edit a current patient's information.

Implementation

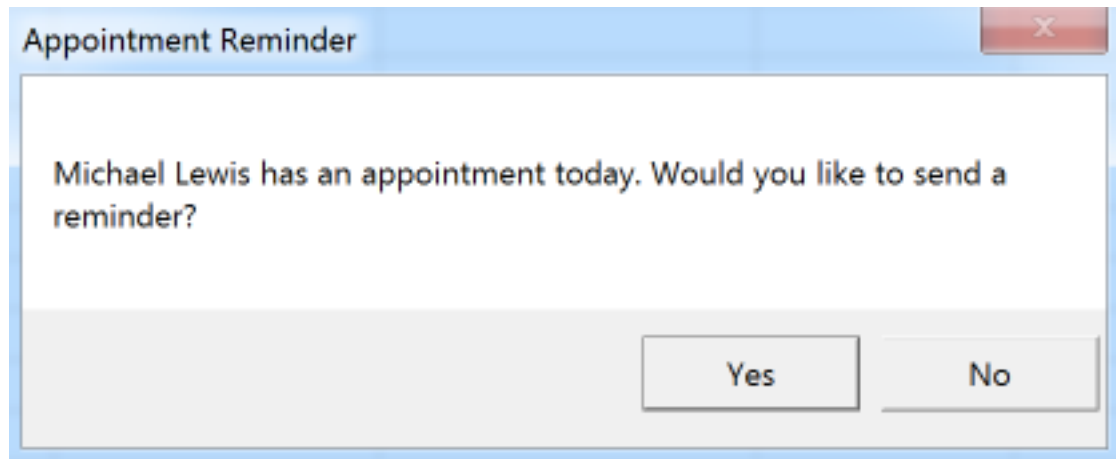
Here is what the spreadsheet looks like

| Patient Appointments | | | | | | | | | |
|----------------------|-----------|------------------|--|------------|---------------|------------------|---------------------|----------------|----------------|
| First Name | Last Name | Appointment Date | Email | Phone | Carrier | Appointment Time | Reminder Requested? | Type Preferred | Reminder Sent? |
| Cameron | Smith | 13-Apr | byuvbata@gmail.com | | US Cellular | 1:15 PM | Yes | Text | No |
| Nathan | Dudley | 14-Apr | byuvbata@gmail.com | | AllTel | 1:30 PM | Yes | Email | No |
| Gove | Allen | 15-Apr | gove@byu.edu | 8013720683 | Sprint | 1:45 PM | Yes | Text | No |
| Michael | Lewis | 12-Apr | michaellewis@fakemail.com | 7029786453 | Verizon | 2:00 PM | Yes | Text | Yes |
| VBA | Test | 17-Apr | vba520test@gmail.com | 1234567890 | AT&T | 2:15 PM | Yes | Text | No |
| VBA | Test | 18-Apr | vba520test@gmail.com | 1234567890 | Sprint | 2:30 PM | No | Email | No |
| VBA | Test | 19-Apr | othervba520test@gmail.com | 1234567890 | T-Mobile | 2:45 PM | Yes | None | No |
| VBA | Test | 20-Apr | othervba520test@gmail.com | 1234567890 | Cricknet | 3:00 PM | No | Text | No |
| John | Doe | 21-Apr | johndoe@fakemail.com | 1234567890 | Boost Mobile | 3:15 PM | Yes | Email | No |
| Jane | Doe | 22-Apr | janedoe@fakemail.com | 1234567890 | Virgin Mobile | 3:30 PM | Yes | Text | No |

Send Reminders Button

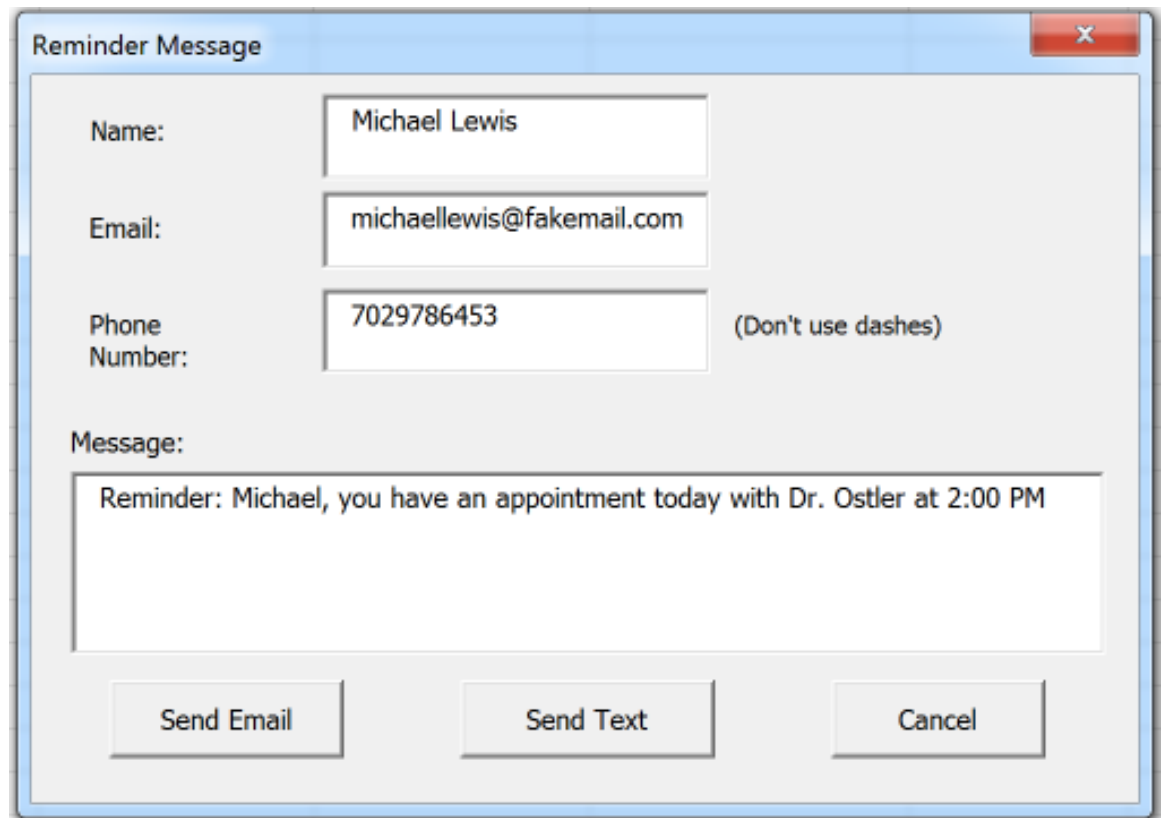


In order to simplify the task of running the macros created, I added the buttons above to the ribbon in Excel. My project has a list of patient information including the patient's next appointment date. When the user clicks the "Send Reminders" button on the ribbon, my code looks to see if any patients have appointments that day and desire reminders to be sent. If a patient does have an appointment and prefers reminders, a box with the patient's name pops up and informs the user that that individual has an appointment that day and asks if the user would like to send a reminder.



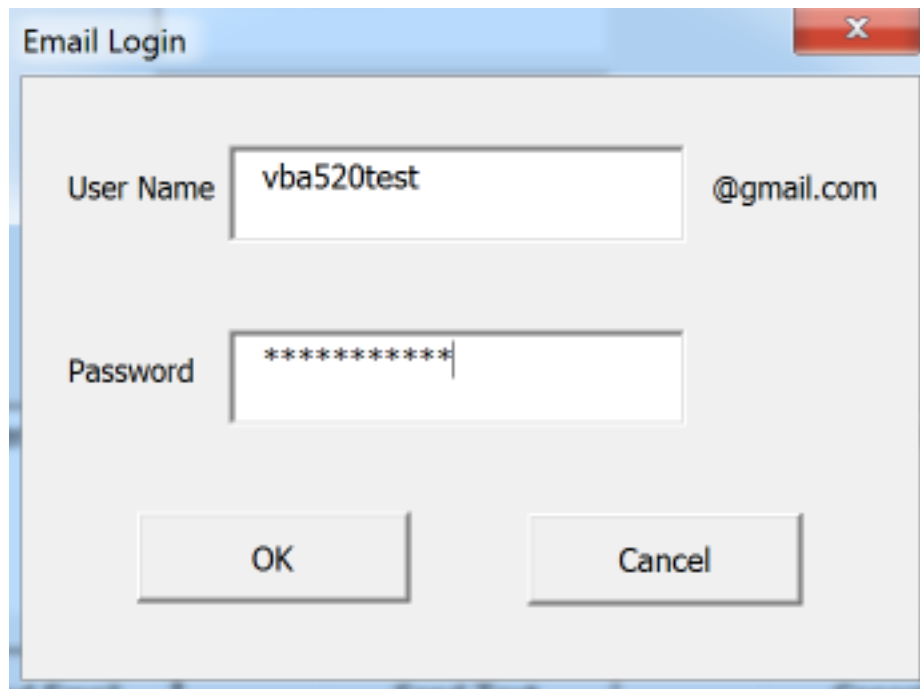
A screenshot of a Windows-style dialog box titled "Appointment Reminder". The dialog has a light blue header bar with a red close button (X) on the right. The main area is white and contains the text "Michael Lewis has an appointment today. Would you like to send a reminder?". At the bottom, there is a light gray bar containing two buttons: "Yes" and "No".

If the user chooses to send a reminder, a user form appears with that patient's information and a pre-customized message already loaded into the form. The user then chooses a type of reminder to send.



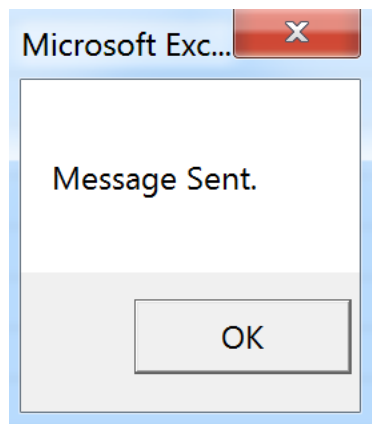
A screenshot of a Windows-style form titled "Reminder Message". The form has a light blue header bar with a red close button (X) on the right. The main area is light gray and contains several input fields and a message box. The fields are labeled "Name:", "Email:", and "Phone Number:". The "Name:" field contains "Michael Lewis". The "Email:" field contains "michaellewis@fakemail.com". The "Phone Number:" field contains "7029786453". To the right of the "Phone Number:" field, there is a note "(Don't use dashes)". Below these fields is a "Message:" label and a text area containing the text "Reminder: Michael, you have an appointment today with Dr. Ostler at 2:00 PM". At the bottom of the form, there are three buttons: "Send Email", "Send Text", and "Cancel".

For the purpose of this assignment, and to protect secure information, I have created another user form that prompts the user with email credentials to send the reminder from. However, when I actually deliver the project, I will have the doctor's email information already integrated into the system and a prompt for credentials will not occur.

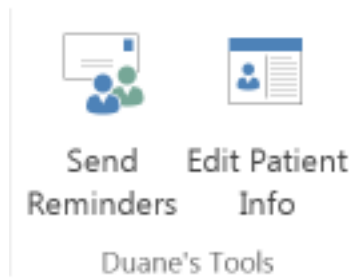


The image shows a standard Windows-style dialog box titled "Email Login". It has a light blue title bar with a red close button (X) on the right. The main area is light gray. There are two input fields: "User Name" and "Password". The "User Name" field contains the text "vba520test" and is followed by "@gmail.com". The "Password" field contains ten asterisks "*****" and has a vertical cursor at the end. Below the input fields are two buttons: "OK" and "Cancel".

Emails and texts can only be sent from email accounts that allow less secure apps to access the account. I created a Gmail account for testing purposes that allows less secure apps to access the account. The account username is: vba520test@gmail.com and the password is vba520test1. Since I have provided these credentials online, I ask anyone who finds this information to not abuse it. Once the user inputs the necessary email credentials, the user will be informed (through a pop-up box) if the reminder was sent.



Edit Patient Info Button

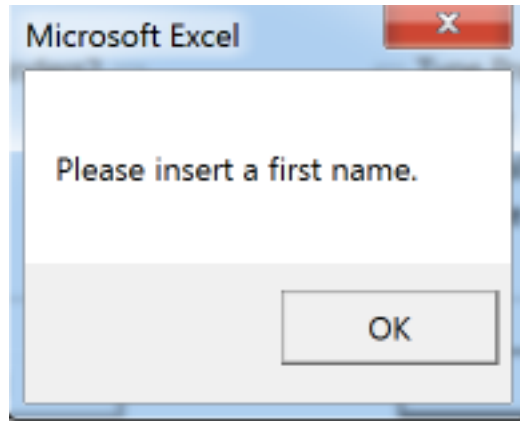


My project also allows the user to add a new patient or edit a current patient's information. The location of the cursor dictates which row is to be edited. To add a new patient, the user puts the cursor in any cell of the row with that patient's information and clicks the "Edit Patient Info" button. To edit a patient's information, the user puts the cursor in any cell of the row with that patient's information and clicks the "Edit Patient Info" button. When the form is brought up, that patient's information will already be loaded onto the form.

A screenshot of a Windows-style dialog box titled "Patient Information". The dialog has a blue title bar with a close button (X) in the top right corner. The main area is light gray and contains four text input fields, each with a label to its left and an asterisk to its right. The fields contain the following text: "First Name:" with "Michael", "Last Name:" with "Lewis", "Phone Number:" with "7029786453", and "Email Address:" with "michaellewis@fakemail.com". Below these fields are two groups of radio buttons. The first group is labeled "Prefers Reminders?" and has two options: "Yes" (selected) and "No". The second group is labeled "Type Preferred" and has three options: "Text" (selected), "Email", and "None". At the bottom of the dialog are two buttons: "Cancel" on the left and "OK" on the right.

The user then edits the information on the form as desired and clicks OK. The information on the spreadsheet will then appear as it was on the user form.

If the user does not enter a patient's first name, last name, or phone number, and tries to hit OK, the user will be notified that there is no value for that data and will be instructed to enter a value. Those three fields are required, as is indicated with a "*" beside each respective input area on the user form.



If the user clicks "No" in the "Prefers Reminders?" area of the form, the "Type Preferred" area will automatically select "None". The purpose of this is to simplify the user's experience on the Edit Patient Info form.

Whenever the user successfully adds a new patient or edits existing patient info, the "Reminder Sent" column on the spreadsheet is populated with the word "No", as requested by the doctor for whom this project is being made.

Learning

My biggest takeaway from this project is how to create a useful project from start to finish. Many of our homework assignments or class projects dealt with one part of a project. On those assignments, everything was already in place to work correctly once we entered the correct code. With my own project, there was no guarantee that everything would work once I put the correct code in one section. I had to learn how to organize my modules and forms so that they would interact with each other correctly. I had to learn how to modify the code I had written in class to make it work for my project. Getting the text messages to use the correct carrier in order to send the message was a little tricky, but I took an idea used in class and applied it to my VBA code to solve that dilemma.

It was important to me for the spreadsheet to be easy to use since the recipient of this project is not familiar with VBA. I tried to incorporate ideas that would minimize the chance of a user changing something that would cause the macros to no longer work. My biggest surprise was that I wanted to keep increasing the scope of my project as I developed it more and more. After I figured out how to do something, I found myself wanting to take it a step further.

Assistance

The only assistance I had on this project was indirect. I used the files from class examples throughout the semester as a guide to my code. There were many instances that I took

code from class examples and manipulated it to work for my project. I also received help from the class TA on one occasion regarding a debugging issue. Other than that, I had no help on this project.