Project Writeup

360 FEEDBACK SPLITTER

Created By:

Ben Geilman

EXECUTIVE SUMMARY

My department in the Business Career Center administers both the Field Study and On-Campus Internship programs for the Marriott School of Management. These are mirror programs that provide real-world experience for students working on teams to solve business problems for contracted companies. The projects have combined participation of roughly 150-200 students per semester.

Student grades are determined primarily by two elements: the feedback of their host company and the feedback of their teammates. This feedback is gathered using Qualtrics surveys that are sent out twice per semester. After the survey is administered, student employees in my office menially separate each student's survey feedback into individual documents and email them to the students for their review.

Splitting and emailing the feedback to the students is extremely time consuming and does not produce attractive reports. Both of these challenges can be solved through VBA automation.

My final product is a macro-enabled spreadsheet template that effectively solves both of the issues in the preceding paragraph. The spreadsheet both performs a complex splitting procedure to produce attractive, individualized reports for each student and emails this custom report to each student.

This new process will eliminate roughly 15 hours of labor time in my role each semester, totaling around $300 of savings per year.

PROJECT IMPLEMENTATION

Rather than explain the process I went through to create my final project and explaining every step of my process in creating it, I will describe my implementation by explaining the user experience as they interact with my final product.

Upon opening a new spreadsheet from my template, the user is presented with a blank sheet with the headings “Student Name”, “Email”, and “Sent?” and a dialog box instructing the user to copy the names and emails of the set of students to receive feedback reports. After copying the emails to the sheet, the user is directed to a custom tab on the ribbon, where two buttons allow them to split the feedback reports or email the individual reports.

Upon clicking either of these two buttons, the user is presented a form. If the Split button is clicked, the user form prompts the user for an input folder location where the raw reports from Qualtrics are stored and an output folder location where they would like the individual reports to be stored.

The user has the option to check a box on this form to automatically email the feedback to the students once the splitting process has completed. If this box is checked, a second dialog box appears prompting the user for an email subject and email body text. Once the user clicks the Split and Email button at the bottom of this form, the Splitting process creates the individual reports using a universal format, followed by the Email process creating the email object, attaching the feedback, and sending each student their personalized feedback report.

In addition to the base processes, I also built several useful error handling features into the code. Naturally, I made sure to create check features in the forms to ensure they are filled out correctly and their folder selections are valid. I also built two primary features into the email process to help catch human error issues and correct them. First, the after an email is sent, the word “Sent” appears beside the name and email address of the student it was sent to. Second, at the end of the process the user is given a dialogue box that lists names of students whose email addresses were not included on the sheet. These two features help the user identify missing information. The email script will not send emails to students who have already been sent an email, which makes it easier to run through the list once corrections have been made.

PROJECT & LEARNING OUTCOMES

This project was an excellent opportunity to solidify several topics covered in class. Though I was able to guess my way through much of the midterm with success, I did not feel confident implementing certain concepts such as passing variables to a sub procedure or appropriately using functions within code.

I learned a great deal about coding discriminating conditions that do not easily produce errors. I encountered several different types of errors as I coded the batch processing of the input documents. These issues dealt with code I wrote to determine where each student's feedback started and ended, when the code reaches the end of valuable information in each document, and which lines in the individual report need special formatting. The majority of my time was spent solving these errors so that the entire process would run bug-free even if every file were not exactly uniform.

I learned a great deal about several objects and properties I had not yet encountered in VBA. The report required me to learn how to create and manipulate charts and make changes to the page setup of a document. I learned how to automatically output a worksheet as a .pdf document.

Through persistence, good teaching, and Google, I was able to accomplish everything with this project that I hoped. The user interface is as simple and intuitive as I had hoped, the process runs as smoothly as I hoped, and the output is as attractive and useful as I had hoped. With all of my goals accomplished, I feel very satisfied with the project. I am even looking forward to other features I could add in the future, such as an email template storage method, or a built-in grading method.

My boss and coworker have been impressed by the outcome of the project, and I am looking forward to implement it for the first time this week. This spreadsheet will likely save 15 hours of work each semester, creating wage cost savings of roughly $300 per year. This time savings can either be used to save money, or to reinvest in other useful projects.