Matt Kimmel

SDEV 140

05 May 2022

Final Project – Validation Testing

***Data Sets***

For my program there are two entry field categories that require input validation. These categories include the quantity entry field and the cost entry field. These fields are necessary for calculating the total cost, so it’s important that no errors are encountered. The quantity field must contain only integers, no letters, and no symbols. The cost field must also contain integers and no letters, but in addition it needs to allow exactly 1 decimal. There are many ways to do this, but I decided to design it so that only the accepted characters can be inputted. If the user tries to type in anything else, it simply won’t show up. There is one other stipulation to be considered, which is an empty string. For the entry fields to work correctly, they must also allow an empty string. Otherwise, the user wouldn’t be able to erase an input. However, an empty string will cause an error if it is used when calculating the total. To work around this, I created an error message box that will pop up if the user clicks the calculate button while one of the entry fields contains an empty string. It will also display which row the empty string is in.

***Development***

When I first began development on input validation, I started with only 2 methods. The first one I called qValid and was used with the quantity to field to allow only integers. This one worked immediately with no issues. The second one I called cValid and was used with the cost field to allow integers and 1 decimal. However, the code I started with did not produce the results I needed. Originally, I just made a copy of the qValid method and added the following line of code:

elif input == ".":

                return True

My hope was that this would allow a decimal in addition to integers, which it did but it also allowed multiple decimals. It took me some time to figure out a way around this, testing out different lines of code and new methods. Finally, I came up with a third method that could be called in the cValid method that checked if the entry was a float. This worked perfectly and so I was able to move on to my last method for input validation.

My last method I called errPopup which contains very simple code that displays a message box if the calculate button is clicked when there is an empty string in one of the entry fields. At first, I thought that I might just write some code that will automatically change any empty strings to a 0, but this assumes that the user intentionally left a field blank because they decided not to use it. However, if this was unintentional then the calculation could be off. So instead, I thought it better to simply inform the user of the issue. The user can then close the message box, correct the error, and click calculate again.

***Test results***

Here are some screenshots of my good test data working. The first screenshot shows a test where I added a line of code to print all keystrokes to the terminal regardless of whether they were accepted or not. This shows that even though letters and symbols were typed, only integers were allowed:

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

This next screenshot shows the message box that is displayed when the user tries to click calculate while an empty string is left in one of the entry fields:

Graphical user interface

Description automatically generated