

Evaluation

- **Evaluation of Key Requirements of the Program (Previously stated in Analysis)**
 - User should be able to input order details with table number and a string of numbers corresponding to menu items chosen
 - ✓ When the user wants to enter an order, the program presents the user with a menu that is able to display a variety of menu items with their respective prices and what categories they are in
 - ✓ The user can then enter an input that contains a list of menu item index number that are separated by commas
 - ✓ The program smart enough to accept input with any amount of spaces between the commas and will reject any input that isn't valid that has invalid characters
 - ✓ The input is validated with a wide range of conditions that are built to reject any type of errors that the user can make when entering input
 - ✓ Allows for a single menu order to be entered and processed in a single line for efficiency
 - ✓ Takes the table number, name of items and how much of each item
 - ✓ This part of the program was built to have thorough validation checks so that the end user could not easily break and crash the program and thus not being user friendly
 - ✓ The program exceeds the expectations of the this requirement
 - ✗ In my opinion, this method of entering menu input is efficient, however unintuitive and could be hard to understand for the end user.
 - Program must be able to validate input data
 - ✓ For every place in the program where the user can enter an input, there has been thorough validation checks
 - ✓ This is so that the user cannot easily break the program and lose their progress and have to start over again
 - ✓ This makes it user friendly and easy to use
 - ✓ Debugging of the validation checks can be viewed in the Debugging file
 - ✓ Exceeds the expectations of the requirements
 - Display the order details for printing
 - ✓ If the user tries to finalize the order and print their order for printing without having ordered anything in the first place, the program will prevent them from doing so
 - ➔ This is so that the user cannot break the program and this ensures that proper information is printed out when finalizing their order
 - ✓ The program will print out a formatted version of the user's final order
 - ✓ It will display how many of each item has been ordered as well as the price totals and the quantity totals of menu items ordered
 - ✓ It will also display what Table ordered the items
 - ✓ At the bottom of the receipt, the program asks the user whether they would like to exit and enter another order, exit and keep current order, or just quit the program
 - ➔ This is so that the user can easily navigate around the program and doesn't have to quit program and execute it again if they want to enter another order
 - ➔ User friendly
 - ✓ Exceeds the expectations of this requirement
 - Loop for next order
 - ✓ After the user has finalized their order, the user can then choose to exit to the main menu and access all of the other features of the program including being able to loop for another order
 - ✓ Allowing the user to exit after finalizing the program and putting them back into the main menu for them to order another item is intuitive and user friendly
 - ✓ This makes it easier for the user to use the program and makes it less difficult to use
 - ✓ The program has accomplished this effectively

- Allow menu to be saved to a file
 - ✓ Everytime the user makes any changes to the menu whether it be editing an element of a menu item (e.g. price, name, or index number), deleting a menu item or adding a menu item, the program will always save the menu file to a file called menu.txt
 - ✓ The program saves it as a menu.txt file in the same directory as main.py
 - ➔ menu.txt contains information about each menu item on each line
 - ➔ Each line is a single menu item
 - ➔ The menu items are in the format of comma separated values
 1. **The first value [0] on a line is the menu index number**
 2. **The second value [1] is the price**
 3. **The third value [2] is the name of the item**
 4. **The fourth value [3] is the category for the item**
 - ✓ ***Upon running main.py for the first time, the program will automatically create menu.txt if it hasn't already been created, and the running_totals.txt as well***
 - ➔ This is so that the program is portable and can be used anywhere
 - ➔ If the user doesn't have the menu file it's not a problem because the program will just generate it itself
 - ✓ The program has accomplished this effectively
- Allow for user to amend, add, delete menu items as well as save menu changes
 - ✓ The user can go into the editing main menu and choose to Add, Edit or Delete menu items
 - ✓ The user can select these menu options where they will be displayed a menu with options based on their selection
 - ➔ Adding a menu item allows the user to add a category and is validated to make sure the input is valid
 - ➔ It checks to make sure the name only contains letters and the first letter of the menu item name is capitalized
 - ➔ Also checks to make sure that the price is a float only.
 - ➔ The price is then displayed to 2 decimal places and is rounded
 - ➔ If price's last digit is a zero, the menu will still display it to 2 decimal places like a real menu
 - ➔ **E.g. "\$2.5" or "\$2.50" would still display as \$2.50**
 - ➔ Editing a menu item:
 - ➔ Allows the user to edit the Index, Name, Price and Category
 - ➔ If the user changes the Index the category will change as well (same applies for the category)
 - ✓ Exceeds the expectations of the requirements
- Maintain running total of order values
 - ✓ Once the program takes in the user's input, it stores it in a variable
 - ✓ Meets the expectations of the requirements
- Maintain running totals of the quantity
 - ✓ Stores it in a file as well like running total
 - ✓ Meets the expectations of the requirements
- Save running totals to a file
 - ✓ Meets the expectations of the requirements
- Provide options to display the menu and running totals
 - ✓ Whenever the use wants to order something, the menu is printed on the screen for the user to see
 - ✓ The menu items all have categories with prices and names displayed in order
 - ✓ The menu displayed with the categories makes it easy to view
 - ➔ I also programmed the menu myself so that based on how long the menu item with the most characters calculates the right amount of periods for each menu item
 - ✓ Exceeds the expectations of the requirements

