UI and Functional test cases for this Tip Calculator

UI Test Cases:

1. **Required Fields:** All the fields should be labeled correctly and should have a textbox to input the values.
2. **Data Type Errors:** If the textbox contains dates, numeric, currency or other specified data types, ensure that only valid data can be entered.
3. **Field Widths:** If the textbox contains text boxes that allow data entry, ensure that the width of data entered does not exceed the width of the table field.
4. **Additional Buttons**: There should be some additional buttons to increase or decrease the values within one click.
5. **Minimise and Maximise buttons:** There should be a minimise button but the maximise button is optional. In Calculators, there is no need for a maximise button.
6. **Grammar and Spelling:** Ensure that you have test cases that look for grammar or spelling errors.
7. **A line of separation**: There should be a line to separate input and output fields.
8. **Proper Use of Symbols:** There should be proper use of symbols for each entry. For example: In tip% box, there should be a '%' symbol and in the output fields there should be a '$' symbol.
9. **Proper Alignment:** All fields on a page (For Example, text box, radio options, drop-down lists) should be aligned properly.
10. **Center Screen opening:** The application should by default open at the centre of the screen.
11. **Suitable Icon**: The application should have a proper icon that justifies its name and function.
12. **Default Values:** The entries can have a default numerical value.
13. **No additional fields:** There should be no additional buttons or boxes. The design should be the same as it was given.
14. **Size:** The size of the interface should be as much as required. No need to have a bigger size than required. Smaller size can lead to some difficulties in using the application.
15. **Typography:** The text should be easily readable. It should have enough contrast with the background.

Functional Test Cases:

1. **Accept Proper Input:** The textboxes should accept the proper input for that particular field. If any field requires only integer input, then all the other characters should be ignored or handled properly.
2. **Proper Boundaries of Inputs:** While using buttons, it should be ensured that those fields which require positive integers, the button which decrements the value should not decrement after 0.

3. **Immediate output:** After entering all the required fields, the application should give immediate output without using any other button.
4. **Decimal Values Precision:** In case of float or double values, it should be ensured that the values will round off upto two decimal places.
5. **On click Output:** If any invalid entry is entered, in that case, the user can click anywhere on the application to get the result after processing the input.
6. **Application crash:** The application should not get crashed for bigger values.
7. **Divide by zero error:** Divide by zero errors should be handled for any calculations.
8. **Tab Button:** Tab and Shift+Tab order should work properly.
9. **Window Size:** Check if the default window size is correct.
10. **Calculation time**: Check if the calculation time is less or not.

In case you change the bill value, click anywhere outside the text box to get the result.