Test Plan for LoanPro Calculator

by @Ricardo Santillan

1. Introduction

1.1 Purpose:
The purpose of this test plan is to outline the strategy for testing the LoanPro calculator application to ensure its functionality, performance, and robustness across various scenarios. This plan covers different types of testing to verify the correctness of arithmetic operations: addition, subtraction, multiplication, and division.

1.2 Scope:
The scope of this testing plan includes functional testing, boundary testing, error handling, performance testing, and usability testing. The focus will be on validating basic arithmetic operations and ensuring compliance with expected behaviors and constraints.

1.3 References:
- Application Requirements Document
- Functional Specifications Document
- Known Bugs and Issues List

2. Test Objectives

- Verify that basic arithmetic operations (add, subtract, multiply, divide) function correctly.
- Ensure that the application handles edge cases and boundary conditions properly.
- Validate that error messages are displayed correctly for unsupported operations and invalid inputs.
- Confirm that the calculator adheres to the rounding rules and scientific notation expectations.
- Test performance under normal and high load conditions.

3. Test Items

- Basic arithmetic operations:
  - Addition
  - Subtraction
  - Multiplication
  - Division
- Error handling for invalid operations and inputs
- Handling of large numbers and scientific notation
- Performance with large operands

4. Test Approach

4.1 Functional Testing:
Test each arithmetic operation with a variety of inputs to ensure correct functionality.

4.2 Boundary Testing:
Test edge cases related to large numbers, small numbers, and results close to rounding boundaries.

4.3 Error Handling:
Verify that the calculator handles division by zero, invalid operand counts, and unsupported results properly.

4.4 Performance Testing:
Evaluate the calculator’s performance with very large numbers and multiple operations to ensure stability and efficiency.

4.5 Usability Testing:
Check the clarity and correctness of user interface elements and error messages.

5. Test Cases

5.1 Addition Operation

Test Case ID	Description	Input	Expected Result
TC_ADD_001	Basic addition with positive integers	3, 5	8
TC_ADD_002	Addition with negative numbers	-4, 6	2
TC_ADD_003	Addition with decimal numbers	1.234, 3.456	4.690
TC_ADD_004	Addition with large numbers	1.0E10, 3.0E10	4.0E10
TC_ADD_005	Addition with small numbers	1.0E-10 3.0E-10	4.0E-10
TC_ADD_006	Addition with rounding result	1.0000001, 1.0000001	2.0000002
TC_ADD_007	Addition with written numbers	One two	Error message
TC_ADD_008	Addition with first operand empty	1, _	Error message
TC_ADD_009	Addition with second operand empty	_, 1	Error message
TC_ADD_010	Addition with null value	Null 2	Error message
TC_ADD_011	Addition with undefined value	1 undefined	Error message

5.2 Subtraction Operation

Test Case ID	Description	Input	Expected Result
TC_SUB_001	Basic subtraction with positive integers	8, 3	5
TC_SUB_002	Subtraction with negative numbers	-3, - 5	2
TC_SUB_003	Subtraction resulting in decimal	5.5, - 2.2	7.7
TC_SUB_004	Subtraction with large numbers	5.0E6, 1.0E6	4.0E6
TC_SUB_005	Subtraction with small numbers	1.0E-8 3.0E-6	-3.23E-7

TC_SUB_006	Subtraction with written numbers	One two	Error message
TC_SUB_007	Subtraction with first operand empty	1, _	Error message
TC_SUB_008	Subtraction with second operand empty	_, 1	Error message
TC_SUB_009	Subtraction with null value	Null 2	Error message
TC_SUB_010	Subtraction with undefined value	1 undefined	Error message

5.3 Multiplication Operation

Test Case ID	Description	Input	Expected Result
TC_MUL_001	Basic multiplication with positive integers	4, 3	12
TC_MUL_002	Multiplication with negative numbers	-4, 5	-20
TC_MUL_003	Multiplication resulting in decimal	1.5, 2.2	3.3
TC_MUL_004	Multiplication with large numbers	1.0E3, 1.0E4	1.0E7
TC_MUL_005	Multiplication with small numbers	1.0E-4, 1.0E-5	0
TC_MUL_006	Multiplication with written numbers	One two	Error message
TC_MUL_007	Multiplication with first operand empty	1, _	Error message
TC_MUL_008	Multiplication with second operand empty	_, 1	Error message
TC_MUL_009	Multiplication with null value	Null 2	Error message
TC_MUL_010	Multiplication with undefined value	1 undefined	Error message

5.4 Division Operation

Test Case ID	Description	Input	Expected Result
TC_DIV_001	Basic division with positive integers	10, 2	5
TC_DIV_002	Division with negative numbers	-10, 2	-5
TC_DIV_003	Division resulting in decimal	7.5, 3.2	2.34375
TC_DIV_004	Division with large numbers	1.0E8, 1.0E6	100
TC_DIV_005	Division with small numbers	1.0E-4, 1.0E-5	10
TC_DIV_006	Division with written numbers	One two	Error message
TC_DIV_007	Division with first operand empty	1, _	Error message
TC_DIV_008	Division with second operand empty	_, 1	Error message

TC_DIV_009	Division with null value	Null 2	Error message
TC_DIV_010	Division with undefined value	1 undefined	Error message

5.5 Error Handling

Test Case ID	Description	Input	Expected Result
TC_ERR_001	Handling division by zero	10, 0	Error: Cannot divide by zero
TC_ERR_002	Handling invalid operand count	5 + 3 + 2	Error message
TC_ERR_003	Handling unsupported results (Infinity/NaN)|	Sqrt -1	Error message

5.6 Performance Testing

Test Case ID	Description	Input	Expected Result
TC_PERF_001	Performance with maximum value	1.0E308 * 1.0E308	Check stability and performance

5.7 Usability Testing

Test Case ID	Description	Input	Expected Result
TC_USAB_001	Verify error messages and UI elements	Any invalid operation	User-friendly error messages and UI clarity

6. Test Execution

6.1 Test Environment:
- Hardware: Standard testing machine with required specifications
- Software: Latest version of LoanPro calculator
- Tools: Automated testing framework, if applicable

6.2 Test Data:
- Predefined input values for various test cases
- Scripts for automated tests

6.3 Execution Schedule:
- Functional testing: [08/07/2024] to [08/07/2024]
- Boundary and error handling testing: [08/07/2024] to [08/07/2024]
- Performance testing: [08/07/2024] to [08/07/2024]

- Usability testing: [08/07/2024] to [08/07/2024]

7. Risks and Mitigations

7.1 Risks:
- Issues with test automation scripts
- Unanticipated bugs in the application

7.2 Mitigations:
- Regular updates and reviews of test scripts
- Collaboration with developers for bug fixes and adjustments

8. Reporting and Review

8.1 Test Reporting:
- Daily status reports during testing phases
- Final test report including test results, issues found, and overall assessment

8.2 Review Process:
- Review test results with stakeholders
- Address any critical issues before release

Bugs Report for LoanPro Calculator

by @Ricardo Santillan

<table>
<tr><th>Test Case Affected</th><th>Bug description</th></tr>
<tr><td>TC_ADD_005
TC_SUB_005</td><td>Substract and adding of small numbers is not working as expected.
<table>
<tr><th>Operation</th><th>Input</th><th>Output</th><th>Expected</th></tr>
<tr><td>substract</td><td>1.0E-8, 3.0E-6</td><td>-0.00000299</td><td>-3.23E7</td></tr>
<tr><td>add</td><td>1.0E-10, 3.0E-10</td><td>0</td><td>4.0E-10</td></tr>
</table></td></tr>
<tr><td>TC_ERR_002</td><td>Error in the adding of 3 numbers, was expected a usability message but I got the sum of the first numbers
<table>
<tr><th>Operation</th><th>Input</th><th>Output</th><th>Expected</th></tr>
<tr><td>add</td><td>5, 3, 2</td><td>8</td><td>Usability message.</td></tr>
</table></td></tr>
<tr><td>TC_ERR_003</td><td>When I use sqrt 5 I´m expecting the unknown operation message, but instead of that I got the usability one.
<table>
<tr><th>Operation</th><th>Input</th><th>Output</th><th>Expected</th></tr>
<tr><td>sqrt</td><td>1</td><td>Usage message</td><td>Error message indicating the operations supported.</td></tr>
</table></td></tr>
<tr><td>TC_ADD_004
TC_SUB_004
TC_MUL_004</td><td>Result consistency. When I use numbers like 1.0E5 and 5.0E4 in multiplication I expect a result in the same format.
<table>
<tr><th>Operation</th><th>Input</th><th>Output</th><th>Expected</th></tr>
<tr><td>substract</td><td>5.0E6, 1.0E6</td><td>4000000</td><td>4.0E10</td></tr>
<tr><td>add</td><td>1.0E10, 3.0E10</td><td>4000000</td><td>4.0E10</td></tr>
<tr><td>multiply</td><td>1.0E3, 1.0E4</td><td>10000000</td><td>1.0E7</td></tr>
</table></td></tr>
</table>

Recommendations:

Error Messages: Use consistent and clear error messages across all operations and try to make the error messages are user-friendly and easy to understand.

Consider the following:

-For the cases were one of the numbers is not valid, the message should be like:
“Error: The [first|second] number you entered is not valid. Please enter a proper number.”

-For the cases were one of the numbers is missing, the message should be like:
“Error: The [first|second] number you entered is missing. Follow the format like: operation number1 number2”