Final SRS Document (Refined Version)

Functional Requirements

1. The program shall execute BasicML instructions as defined in the specification.
2. The program shall provide a GUI that allows users to navigate available functions.
3. The program shall validate user input, ensuring only integer values are accepted where required.
4. The program shall store up to 100 words in memory at a time.
5. The program shall include an accumulator to store intermediate results of operations.
6. The program shall allow users to input data via keyboard or file upload.
7. The simulator shall detect and display the following errors:
    - Invalid Instructions (Unrecognized opcodes)
    - Memory Overflow (Exceeding 100 memory slots)
    - Divide-by-Zero Errors (Prevent division by zero operations)
8. The program shall display execution results through the GUI output window.
9. The simulator shall allow users to search for specific values in memory.
10. The program's GUI shall display the contents of memory in real-time.
11. The GUI shall include tooltips or a Help menu explaining available commands and functionalities.
12. The program shall support execution of the following arithmetic operations:

- ADD (Addition)
- SUBTRACT (Subtraction)
- MULTIPLY (Multiplication)
- DIVIDE (Division)

13. The simulator shall maintain a log of executed commands that users can view.
14. The program shall halt execution when encountering the HALT (`4300`) instruction.
15. The program shall update and display the accumulator and instruction counter in real-time.

Non-Functional Requirements

1. The program's GUI layout shall follow standard usability principles, ensuring clear navigation with labeled buttons and tooltips.
2. The simulator shall handle invalid input by:
    - Preventing crashes through error handling.
    - Displaying clear, descriptive error messages.
3. Each instruction shall be executed in under 100 ms to ensure smooth user interaction.