Functional Requirements

1. **Memory Allocation:** The system shall provide a memory space of 100 words, where each word is stored as a signed four-digit decimal number.

2. **Program Loading:** The system shall allow users to load a BasicML program into memory beginning at location 00.

3. **READ Instruction:** When a READ (10) instruction is encountered, the system shall prompt the user for a valid signed four-digit decimal input and store the value at the specified memory location.

4. **WRITE Instruction:** When a WRITE (11) instruction is executed, the system shall retrieve the value from the specified memory location and display it on the screen.

5. **LOAD Instruction:** When a LOAD (20) instruction is executed, the system shall copy the value from the specified memory location into the accumulator.

6. **STORE Instruction:** When a STORE (21) instruction is executed, the system shall copy the value from the accumulator into the specified memory location.

7. **ADD Instruction:** When an ADD (30) instruction is executed, the system shall add the value from the specified memory location to the accumulator and update the accumulator with the result.

8. **SUBTRACT Instruction:** When a SUBTRACT (31) instruction is executed, the system shall subtract the value from the specified memory location from the accumulator and update the accumulator with the result.

9. **DIVIDE Instruction:** When a DIVIDE (32) instruction is executed, the system shall divide the accumulator by the value from the specified memory location, update the accumulator with the result, and if a division by zero is attempted, the system shall halt execution and display an appropriate error message.

10. **MULTIPLY Instruction:** When a MULTIPLY (33) instruction is executed, the system shall multiply the accumulator by the value from the specified memory location and update the accumulator with the result.

11. **BRANCH Instruction:** When a BRANCH (40) instruction is executed, the system shall set the execution pointer to the specified memory location.

12. **BRANCHNEG Instruction:** When a BRANCHNEG (41) instruction is executed, the system shall set the execution pointer to the specified memory location if the accumulator contains a negative value.

13. **BRANCHZERO Instruction:** When a BRANCHZERO (42) instruction is executed, the system shall set the execution pointer to the specified memory location if the accumulator contains a zero value.

14. **HALT Instruction:** When a HALT (43) instruction is executed, the system shall immediately stop program execution.

15. **Execution Logging:** The system shall log each executed instruction and all memory changes in a log file or display panel, with each log entry including a timestamp, the instruction executed, and the resulting state of memory.

Non-Functional Requirements

1. **Performance:** The system shall execute a full BasicML program (up to 100 instructions) within 0.5 seconds from the moment the user clicks the "Run" button, under typical workload conditions.

2. **Usability:** The system's graphical user interface (GUI) shall:
    a. Provide buttons with clear and unique labels.
    b. Present a layout verified by a usability study to be "clear" (e.g., at least 80% of test users confirm ease of navigation).
    c. Include tooltips for each interactive element that accurately describe its functionality.
3. **Maintainability and Documentation:** The project shall include documentation for all functions and classes and adhere to industry-standard coding conventions. Additionally, the system's architecture shall ensure that modules are loosely coupled to facilitate easy maintenance and future enhancements.