1. **Memory Initialization**: The system shall initialize with a memory size of 100 words, each capable of storing a signed four-digit decimal number ranging from -9999 to +9999.
2. **Program Loading**: The system shall load a BasicML program from a .txt file into memory, starting at memory location 00.
3. **Instruction Format Interpretation**: The system shall interpret each BasicML word using the first two digits as the opcode and the last two digits as the operand.
4. **READ Instruction**: The system shall read a word from the keyboard and store it in a specified memory location when executing the READ instruction with opcode 10.
5. **WRITE Instruction**: The system shall write a word from a specified memory location to the screen when executing the WRITE instruction with opcode 11.
6. **LOAD Instruction**: The system shall load a word from a specified memory location into the accumulator when executing the LOAD instruction with opcode 20.
7. **STORE Instruction**: The system shall store the word in the accumulator into a specified memory location when executing the STORE instruction with opcode 21.
8. **ADD Instruction**: The system shall add a word from a specified memory location to the accumulator and handle overflow by wrapping around when executing the ADD instruction with opcode 30.
9. **SUBTRACT Instruction**: The system shall subtract a word from a specified memory location from the accumulator and handle underflow by wrapping around when executing the SUBTRACT instruction with opcode 31.
10. **BRANCH Instruction**: The system shall jump to a specified memory location unconditionally when executing the BRANCH instruction with opcode 40.
11. **BRANCHNEG Instruction**: The system shall jump to a specified memory location if the accumulator is negative when executing the BRANCHNEG instruction with opcode 41.
12. **HALT Instruction**: The system shall stop the execution of the program when executing the HALT instruction with opcode 43.

#### **3. Non-Functional Requirements**

1. **User-Friendly Interface**: The system shall provide a user-friendly interface that allows easy interaction for entering data and viewing memory contents, suitable for educational purposes.
2. **Performance**: The system shall execute instructions and respond to user inputs promptly, ensuring efficient and effective operation.
3. **Documentation and Testing**: The system shall include comprehensive documentation for both users and developers, detailing usage instructions and system operations, and it shall include automated tests to verify the correct execution of instructions.