**Functional Requirements:**

1. Load Programs: The UVSim should let users load BasicML programs into the computer's memory from a file.
2. Step-by-Step Execution: The UVSim should run the BasicML programs step by step, starting at memory location 0.
3. Input and Output: The UVSim should allow users to type in values from the keyboard and display output on the GUI while the program is running.
4. Memory Management: The UVSim should handle the computer's memory, letting users store and get data values and program instructions from specific memory locations.
5. Math Operations: The UVSim should do math operations like addition, subtraction, multiplication, and division so that users can work with numbers in memory.
6. Decision Making: The UVSim should have instructions that let users change the program's flow based on certain conditions or values in memory. For example, jump to a different current working location in memory.
7. Handling Errors: The UVSim should be able to detect and deal with errors properly. It should display messages to users when they do something wrong.
8. Stopping Programs: The UVSim should allow users to stop a program with the halt command, so they can end it and start over if needed.
9. Program Debugging: The UVSim should provide debugging through the test file. This makes it so that the file can be running as expected, even after updates.
10. Program Counter: The UVSim should keep track of the location counter, displaying the current instruction's location in memory.
11. Instruction Set Expansion: The UVSim should support the addition of new BasicML instructions, allowing users to extend the functionality of the virtual machine beyond the existing operations.
12. Program Storage: The UVSim should provide a mechanism for users to load BasicML programs from external files.
13. Program Visualization: The UVSim should offer a visual representation of the program execution, displaying the current instruction, memory, and accumulator.
14. Accumulator handling: The UVSim should store information in the accumulator that the program can use in later instructions.
15. Error Logging: The UVSim should log errors and exceptions encountered during program execution.

**Non-Functional Requirements:**

1. Easy to Use: UVSim's GUI should be easy to understand and use, with clear buttons and labels so that all people can navigate it without much trouble.
2. Fast Performance: The UVSim should run programs quickly, not taking too long to respond to user actions, so people can see results without waiting a long time.
3. No Crashes: The UVSim should be stable and not crash during use, even if the user makes an error. It should handle it silently if possible.