**Requirements**

Functional

1. The system shall provide a user interface with a memory capacity of 100 words, each word being a 4-digit signed integer.
2. The system shall have an accumulator register to store and manipulate values during execution.
3. The system shall have a program counter to keep track of the current instruction being executed.
4. The system shall validate input instructions to ensure they conform to the required format.
5. The system shall handle and report errors, and then terminate the program
6. The system shall allow the loading of a program from a text file into the virtual machine's memory.
7. The system shall display the current state of the virtual machine, including the program counter, accumulator value, and memory contents.
8. The system shall handle negative values correctly, using a 4-digit representation with a leading minus sign.
9. The system shall handle positive values correctly, using a 4-digit representation with a leading plus sign.
10. The system shall truncate overflow and underflow conditions when performing arithmetic operations.
11. The system shall validate branch addresses to ensure they are within the valid memory range.
12. The system shall implement a halt operation to terminate the execution of the program.
13. The system shall allow the user to load and execute programs from external text files.
14. The system shall allow the user to step through the program execution one instruction at a time, displaying the updated state of the virtual machine after each step.
15. The system shall display the final state of the virtual machine upon termination.

Non-functional

1. The system shall be implemented in Python and follow best coding practices, including modular design and error handling.
2. The system shall have a clear and user-friendly interface, with informative error messages and instructions for use.
3. The system shall be platform-independent and capable of running on different operating systems without modifications.