

# Software Requirement Specification

## Functional Requirements

1. The system shall have a graphical user interface displaying the program counter, accumulator, memory, and output values.
2. The system shall provide a memory capacity of 100 words, each word being a 4-digit signed integer.
3. The system shall have an accumulator register to store and manipulate values during execution.
4. The system shall have a program counter to keep track of the current instruction being executed.
5. The system shall accept and execute the full range of the UVSim instruction set architecture.
6. The system shall allow the user to select a file from their local machine to be processed by the simulator.
7. The system shall load the input file's contents into its memory after a properly formatted file is provided.
8. The system shall implement all basic operations (read, write, load, store, arithmetic, and branching) as specified in the instruction set.
9. The system shall handle negative values using a 4-digit representation with a leading minus sign, and positive values with a leading plus sign.
10. The system shall truncate over- and underflowing integer values before saving them to memory.
11. The system shall validate branch addresses to ensure they are within the valid memory range.

12. The system shall refuse to load a program that lacks at least one HALT instruction to avoid infinite loops.
13. The system shall output a warning message upon receiving invalid user input.
14. The system shall repeat a request for user input until valid input is provided.
15. The system shall display the final state of the virtual machine upon program termination.
16. The system shall display an error message upon reaching an invalid instruction and terminate.

## **Non-Functional Requirements**

1. The system shall be platform-independent and capable of running on Mac, Linux, and Windows operating systems without modifications.
2. The system shall initialize and be available for user input in under three seconds.
3. The system shall be implemented in Python and follow best coding practices, including modular design and error handling.