

## Team 1 SRS Document

### Functional Requirements

1. The system shall allow users to select a text file containing instructions for the UVSIM.
2. The system shall execute read operations, prompting the user to input values for specific memory locations.
3. The system shall execute write operations, displaying the value stored at a specified memory location.
4. The system shall execute load operations, loading values from specific memory locations into the accumulator.
5. The system shall execute store operations, storing the value in the accumulator into specified memory locations.
6. The system shall support basic math operations such as addition, subtraction, multiplication, and division.
7. The system shall correctly evaluate the conditions for branching (negative or zero accumulator) and jump to the specified memory location accordingly.
8. The system shall handle file loading errors, such as invalid file formats or inaccessible files, and in case of invalid input, unsupported operations, or runtime errors it'll notify users by providing informative error messages to users.
9. The system shall provide a GUI for interacting with the simulator, allowing users to input machine language, view program output, and control the program execution
10. The system shall support the execution of programs with varying lengths and complexities, accommodating different input file sizes and instruction sets.
11. The system shall allow users to export the current state of the UVSIM machine, including memory contents and program counter, for analysis or sharing
12. The system shall provide functionality for users to create and save their own UVSIM programs within the application for future use.
13. The system shall allow users to pause, resume, and reset program execution at any point.
14. The system shall provide options for users to step through the program, executing one instruction at a time.
15. The system shall maintain the integrity of program execution by preventing unauthorized modifications to critical components, such as the program counter and accumulator.
16. The system shall log program execution details, including executed instructions and memory changes, for debugging and analysis purposes.

### Non functional requirements:

1. The system's GUI shall be simple with clear instructions on how to interact with it.

2. The system's code shall be written and designed in a way that'll minimize the execution time and provide ease for maintenance.
3. The system shall provide clear and informative error messages to assist users in troubleshooting.
4. The system shall be scalable, capable of handling increasing user loads and expanding functionality without significant performance degradation.