**Software Requirements Specification (SRS) for UVSim**

**Functional Requirements:**

UVSim shall:

1. Execute Basic Machine Language (BML) instructions sequentially.

2. Accept a properly formatted text file as user input and correctly parse and load BML programs.

3. Provide a text area in the GUI where users can input and edit program instructions.

4. Store and display memory addresses and values in a scrollable table, allowing users to inspect and modify memory contents.

5. Implement an Accumulator register to store intermediate values and perform arithmetic operations, updating the display accordingly.

6. Support conditional and unconditional branching instructions based on the Accumulator’s value and update execution flow accordingly.

7. Execute programs when the “Execute” button is pressed, running all instructions until a halt instruction is reached or an error occurs.

8. Provide a “Step” button to execute a single instruction at a time, updating the GUI with each step.

9. Detect and handle errors such as invalid instructions, memory overflows, and division by zero, displaying clear error messages.

10. Enable users to save programs as text files and reload them without modification or data loss.

11. Allow users to manually halt execution at any point via a dedicated “Quit” button.

12. Display the full memory state in an accessible format for user inspection, refreshing after every executed instruction.

13. Show the current Accumulator value dynamically, ensuring real-time updates during execution.

14. Maintain an execution log that records every executed instruction, including memory changes, for debugging purposes.

15. Accept user input when required by specific instructions, processing the input correctly and updating relevant components.

**Non-Functional Requirements**

1. The system shall provide a GUI designed using Kivy, ensuring clear navigation, responsive design, and accessibility.

2. The system shall execute BML programs with an average response time of less than 500 milliseconds per instruction.

3. The system shall handle invalid inputs without crashing, ensuring error messages guide the user toward corrective actions.