Functional:

1. **Program execution** – system shall execute Basic Machine Language instructions sequentially
2. **File Input Handling** – The system shall allow users to load BML programs from text files
3. **Memory Management** – The system shall store and retrieve data in a simulated memory
4. **Accumulator** – The system shall perform arithmetic operations using an accumulator.
5. **Branching and Control Flow** – The system shall support branching based on accumulator values.
6. **Execution Logging –** The system shall maintain a log of executed instructions for debugging and review
7. **Step by step Execution** – The system shall allow users to execute the program pausing between each instruction for review and debugging
8. **User number input handling** – The system shall allow users to input four-digit values when prompted by specific instructions.
9. **Memory inspection** – The system shall allow users to inspect all memory.
10. **Accumulator state inspection** – The system shall display current accumulator value for user inspection.
11. **Program halting** – The system shall allow users to stop execution.
12. **Instruction Execution Control** – The system shall allow users to stop and execute execution.
13. **File saving** – The system shall allow users to save the current state of the program.
14. **Error handling** – The system shall display informative error messages for invalid instructions.
15. **Log displaying** – The system shall allow users to see complete log of all execution.

Non Functional:

1. **Usability** – The system shall provide a GUI that is easy to understand and navigate.
2. **Performance** - The system shall execute Basic Machine Language programs with minimal delay
3. **Reliability** - The system shall handle invalid inputs gracefully and prevent crashes or system failures.