**Functional Requirements:**

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. **User Input Handling** - The GUI shall provide a text area for users to input program instructions.
4. **Memory Management** – The system shall store and retrieve data in a scrollable table that displays addresses and values.
5. **Accumulator** – The system shall perform arithmetic operations using an accumulator.
6. **Branching and Control Flow** – The system shall support branching based on accumulator values.
7. **Execute Button** - A "Execute" button shall execute the loaded program until completion.
8. **Stepping** - A "Step" button shall execute the next instruction and update the GUI.
9. **Error handling** – The system shall display informative error messages for invalid instructions in plain language.
10. **File saving** – The system shall allow users to save the current state of the program.
11. **Program halting** – The system shall allow users to stop execution.
12. **Memory inspection** – The system shall allow users to inspect all memory.
13. **Accumulator state inspection** – The system shall display current accumulator value for user inspection.
14. **Log displaying** – The system shall allow users to see complete log of all execution.
15. **Execution Logging –** The system shall maintain a log of executed instructions for debugging and review

**Non-functional Requirements:**

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.