**Project Name**: UVSim (Universal Virtual Simulator) with GUI  
**Author**: Michael Findlay  
**Date**: 6/12/2025

**Functional Requirements**

1. **The system shall allow the user to load a .txt file containing BasicML code from the local file system.**
2. **The system shall parse and validate each line of the loaded file to ensure it is a valid BasicML instruction or data word.**
3. **The system shall display an error message if a line in the file is malformed or invalid.**
4. **The system shall allow the user to run the loaded BasicML program via a “Run” button.**
5. **The system shall display each input prompt specified by a READ (opcode 10) instruction using a graphical input dialog.**
6. **The system shall capture and store user input as a signed four-digit integer and place it in the correct memory location.**
7. **The system shall execute WRITE (opcode 11) instructions by displaying output in a GUI output area or message box.**
8. **The system shall visually display the current state of the accumulator during execution.**
9. **The system shall execute LOAD and STORE instructions by transferring data between memory and the accumulator.**
10. **The system shall execute arithmetic operations (ADD, SUBTRACT, MULTIPLY, DIVIDE) and update the accumulator accordingly.**
11. **The system shall execute branching instructions (BRANCH, BRANCHNEG, BRANCHZERO) and alter program flow as specified.**
12. **The system shall halt execution when a HALT (opcode 43) instruction is encountered.**
13. **The system shall highlight or indicate the currently executing instruction in the GUI (optional but recommended for user clarity).**
14. **The system shall allow the user to reset the simulator to its initial state via a “Reset” button.**
15. **The system shall display all memory values in a table or grid format, updating as values change during execution.**

**Non-Functional Requirements**

1. **The system shall respond to user input (e.g., file selection, input dialogs) within 1 second.**
2. **The system shall be compatible with Windows operating systems and require .NET 6.0 or higher.**
3. **The system shall provide a user interface that is visually clean, with labels for all controls, and usable without prior training.**